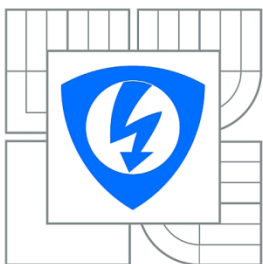




VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH  
TECHNOLOGIÍ

ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF RADIO ELECTRONICS

## UNIVERZÁLNÍ LOGICKÝ ANALYZÁTOR

GENERIC LOGIC ANALYZER

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

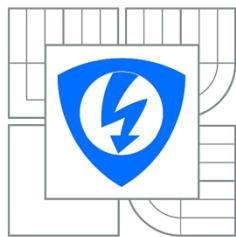
Bc. MARTIN TAJČ

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. MICHAL KUBÍČEK, Ph.D.

BRNO 2013



VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

Ústav radioelektroniky

# Diplomová práce

magisterský navazující studijní obor  
**Elektronika a sdělovací technika**

**Student:** Bc. Martin Tajč  
**Ročník:** 2

**ID:** 119639  
**Akademický rok:** 2012/2013

**NÁZEV TÉMATU:**

## Univerzální logický analyzátor

### POKYNY PRO VYPRACOVÁNÍ:

Navrhněte koncepci rychlého logického analyzátoru založeného na obvodu FPGA. Analyzátor bude schopen snímat stav signálů paralelních sběrnic i sériových rozhraní (SPI, I2C) a provádět jejich dekódování. Realizujte akviziční jednotku. Jako základní modul použijte vývojovou desku s obvodem FPGA, navrhněte a realizujte hardware pro připojení logického analyzátoru k měřenému obvodu. Navrhněte základní bloky firmwaru FPGA.

Vytvořte program pro PC, který bude realizovat uživatelské rozhraní akviziční jednotky. Program bude umožňovat ovládání funkce, interpretaci a ukládání dat.

### DOPORUČENÁ LITERATURA:

[1] MAXFIELD, C. The Design Warrior's Guide to FPGAs. 1st ed. Burlington: Newnes – Elsevier, 2004.

**Termín zadání:** 11.2.2013

**Termín odevzdání:** 24.5.2013

**Vedoucí práce:** Ing. Michal Kubíček, Ph.D.

**Konzultanti diplomové práce:**

**prof. Dr. Ing. Zbyněk Raida**  
*Předseda oborové rady*

### UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## **ABSTRAKT**

Tato diplomová práce je zaměřena na návrh univerzálního logického analyzátoru založeného na obvodu FPGA. Hlavním cílem tohoto analyzátoru je analýza protokolů I<sup>2</sup>C, SPI, RS232, RS485 a vstupní šestnáctibitové sběrnice. Analyzátor obsahuje grafický displej a možnost připojení k osobnímu počítači pro interpretaci naměřených hodnot a ovládání pomocí vytvořené aplikace. První část této práce se zabývá popisem jednotlivých funkčních bloků a popisem použití přímo v daném zařízení. V následující části jsou popsány jednotlivé datové sběrnice, které univerzální logický analyzátor dokáže zpracovat. V závěrečné části je popsán vytvořený subsystém v obvodu FPGA, počítačová aplikace a také zdrojový kód pro mikrokontrolér ATmega32.

## **KLÍČOVÁ SLOVA**

Logický analyzátor, FPGA, mikrokontrolér, LCD, USB, I<sup>2</sup>C, SPI, RS232, RS485

## **ABSTRACT**

The aim of the project is to design a generic logic analyzer based on an FPGA. The analyzer should be able to analyze protocols such as I2C, SPI, RS232, RS485 and GPIO. The captured data can be observed using an embedded graphical display or using a PC. The PC can be used to control the analyzer and to save the captured data. The thesis is divided into several parts. First, the basic structure of the analyzer is described including detailed description of its particular components. Later on, the most common protocols that can be decoded using the analyzer are described. Finally, the FPGA subsystem and microcontroller application are presented together with corresponding source codes.

## **KEYWORDS**

Logic analyzer, FPGA, microcontroller, LCD, USB, I<sup>2</sup>C, SPI, RS232, RS485

TAJČ, M. *Univerzální logický analyzátor*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2013. 72 s. Vedoucí diplomové práce Ing. Michal Kubíček, Ph.D..

## PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma Univerzální logický analyzátor jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne .....

.....

(podpis autora)

## PODĚKOVÁNÍ

Děkuji vedoucímu diplomové práce Ing. Michalu Kubíčkoví, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce. Dále bych chtěl poděkovat své přítelkyni za podporu.

V Brně dne .....

.....

(podpis autora)

Výzkum realizovaný v rámci této diplomové práce byl finančně podpořen projektem  
CZ.1.07/2.3.00/20.0007 **Wireless Communication Teams**  
operačního programu **Vzdělávání pro konkurenceschopnost**.



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Finanční podpora byla poskytnuta Evropským sociálním fondem  
a státním rozpočtem České republiky.

Tento příspěvek vzniknul za podpory projektu CZ.1.07/2.3.00/20.0007 WICOMT,  
financovaného z operačního programu Vzdělávání pro konkurenceschopnost



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

# OBSAH

<b>SEZNAM OBRÁZKŮ .....</b>	<b>ix</b>
<b>SEZNAM TABULEK .....</b>	<b>xi</b>
<b>ÚVOD.....</b>	<b>1</b>
<b>1 POPIS A REALIZACE ANALYZÁTORU .....</b>	<b>2</b>
1.1 POPIS LOGICKÉHO ANALYZÁTORU .....	2
1.2 MOŽNOSTI REALIZACE .....	2
<b>2 POPIS JEDNOTLIVÝCH BLOKŮ A JEJICH REALIZACE.....</b>	<b>4</b>
2.1 NAPÁJECÍ ČÁST .....	5
2.2 IN-16 .....	5
2.3 MĚNIČ NAPĚŤOVÉ ÚROVNĚ SN74ALVC164245.....	6
2.4 FYZICKÉ ROZHRANÍ RS232, RS485 A CAN .....	6
2.5 OBVOD FPGA .....	8
2.6 MIKROKONTROLÉR ATMEGA32 .....	12
2.7 LCD DISPLEJ .....	16
2.8 DOTYKOVÁ FÓLIE.....	18
2.8.1 Kalibrace dotykové fólie.....	20
2.9 VSTUPNÍ BLOKY A/D A IRDA .....	20
2.10 KOMUNIKACE OSOBNÍHO POČÍTAČE S LOGICKÝM ANALYZÁTOREM .....	21
<b>3 POPIS ANALYZOVANÝCH DATOVÝCH PROTOKOLŮ .....</b>	<b>22</b>
3.1 I <sup>2</sup> C.....	23
3.2 SPI.....	24
3.3 RS232 .....	26
3.4 RS485 .....	28
<b>4 VYTVOŘENÝ SUBSYSTÉM V OBVODU FPGA.....</b>	<b>29</b>
4.1 PROGRAMOVÁNÍ OBVODŮ FPGA .....	29
4.2 STRUKTURA SUBSYSTÉMU FPGA.....	29
<b>5 VYTVOŘENÁ POČÍTAČOVÁ APLIKACE.....</b>	<b>33</b>
5.1 HLAVNÍ OVLÁDACÍ MENU .....	33
5.2 GRAFICKÝ NÁVRH LOGICKÉHO ANALYZÁTORU .....	33
5.3 VYTVOŘENÉ FUNKCE LOGICKÉHO ANALYZÁTORU .....	35
5.3.1 Volba snímacího režimu .....	35
5.3.2 Použití kurzorů .....	35
5.3.3 Ovládání snímání a zobrazené oblasti .....	36
5.3.4 Ukládání a nahrávání nasnímaných hodnot .....	36
5.3.5 Vytvoření zdrojového kódu pro mikrokontrolér řady AVR .....	37
5.3.6 Volby zobrazení .....	38
5.3.7 Tisk .....	39
5.4 VYTVOŘENÍ VLASTNÍ SBĚRNICE .....	41
5.5 VOLTMETR .....	41
5.6 O APLIKACI .....	41
5.7 OVLÁDÁNÍ POMOCÍ KLÁVESOVÝCH ZKRATEK .....	42

5.8 KOMUNIKACE OSOBNÍHO POČÍTAČE S VYTVOŘENÝM HARDWAREM LOGICKÉHO ANALYZÁTORU .....	43
<b>6 VYTVOŘENÝ PROGRAM PRO MIKROKONTROLÉR ATMEGA32 .....</b>	<b>47</b>
6.1 PROGRAMOVÁNÍ MIKROKONTROLÉRU .....	47
6.2 ZÁKLADNÍ VÝVOJOVÝ DIAGRAM .....	47
<b>7 POPIS PŘIPOJENÍ ANALYZÁTORU .....</b>	<b>50</b>
7.1 POPIS KONEKTORŮ PRO PŘIPOJENÍ ANALYZOVANÝCH PERIFÉRIÍ .....	51
7.2 POPIS OSTATNÍCH KONEKTORŮ .....	51
<b>8 POROVNÁNÍ DOSAŽENÝCH PARAMETRŮ .....</b>	<b>53</b>
8.1 VYTVOŘENÝ ANALYZÁTOR .....	53
8.2 AXIS SIGMA2 .....	53
8.3 LOGIC ANALYZER SALEAE .....	54
<b>9 ZÁVĚR .....</b>	<b>55</b>
<b>LITERATURA .....</b>	<b>56</b>
<b>SEZNAM ZKRATEK A SYMBOLŮ .....</b>	<b>58</b>
<b>SEZNAM PŘÍLOH .....</b>	<b>60</b>



# SEZNAM OBRÁZKŮ

Obr. 1: Blokové schéma navrhnutého obvodu .....	4
Obr. 2: Optimalizace navrhnutého spínaného zdroje, převzato z [12] .....	5
Obr. 3: Napájecí část obvodu.....	5
Obr. 4: Zapojení obvodu SP3232ECN, převzato z [12] .....	7
Obr. 5: Zapojení obvodu ADM3493, převzato z [13] .....	7
Obr. 6: Zapojení obvodu TJA1040, převzato z [14].....	8
Obr. 7: Vnitřní zapojení obvodu FPGA, převzato z [2] .....	8
Obr. 8: Blokové zapojení vývojového kitu s obvodem FPGA, převzato z [15] .....	9
Obr. 9: Rozložení bloků v obvodu Spartan-3E, převzato z [16] .....	10
Obr. 10: Simulace obvodu pro konverzi z 5 V na 3,3 V .....	11
Obr. 11: Časová simulace obvodu pro konverzi z 5 V na 3,3 V .....	12
Obr. 12: Připojení grafického displeje k mikrokontroléru ATmega32.....	17
Obr. 13: Úvodní obrazovka, hlavní menu a menu funkcí analyzátoru .....	18
Obr. 14: Grafický design pro I <sup>2</sup> C, SPI, RS232, RS485, pro funkce A/D a GPIO .....	18
Obr. 15: Praktická aplikace vytvořeného ovládacího menu .....	18
Obr. 16: Konstrukce dotykové fólie, převzato z [4] .....	19
Obr. 17: Zapojení vývodů dotykové fólie, převzato z [4] .....	19
Obr. 18: Připojení dotykové fólie k mikrokontroléru ATmega32 .....	20
Obr. 19: Schematické zapojení obvodu FT232RL, převzato z [19] .....	21
Obr. 20: Struktura asynchronního rámce pro režim 7E2, převzato z [5].....	22
Obr. 21: Propojení jednotlivých zařízení na sběrnici I <sup>2</sup> C .....	23
Obr. 22: Zobrazení přenosu jednoho datového bytu pře sběrnici I <sup>2</sup> C, převzato z [6] .....	24
Obr. 23: Zobrazení komunikačního paketu sběrnice I2C, převzato z [6].....	24
Obr. 24: propojení jednotlivých zařízení na sběrnici SPI, převzato z [7].....	25
Obr. 25: Různé fáze komunikace pře SPI, převzato z [3] .....	26
Obr. 26: Napěťové úrovně standardu RS232 a TTL, převzato z [5] .....	27
Obr. 27: Zapojení konektoru Cannon 9 .....	27
Obr. 28: Schematické zapojení Balanced differential páru, převzato z [21] .....	28
Obr. 29: Vytvořený subsystém v obvodu FPGA .....	30
Obr. 30: Hlavní menu vytvořené aplikace, aktuální výběr funkce analyzátoru .....	33
Obr. 31: Grafické rozhraní vytvořeného analyzátoru .....	34
Obr. 32: Informace o nepřesném vytvoření nasnímaného signálu .....	38
Obr. 33: Zobrazení nasnímaných hodnot pro režimy UART, RS232, RS485. I <sup>2</sup> C a SPI .....	39
Obr. 34: Ukázka automatické úpravy zobrazení nasnímaného průběhu .....	39
Obr. 35: Zobrazení vytisknutého formuláře .....	40
Obr. 36: Zobrazení aplikace pro měření dvou hodnot napětí a jejich grafický záznam ...	41
Obr. 37: Vytvořený modul pro zobrazení informací o aplikaci.....	42
Obr. 38: Způsob komunikace osobního počítače s analyzátozem .....	44
Obr. 39: Vývojový diagram vytvořeného programu pro mikrokontroléru ATmega32 ....	48
Obr. 40: Rozložení matice pro výběr stisknuté oblasti.....	49
Obr. 41: Diagram větvení vytvořeného menu pro grafický displej .....	49
Obr. 42: Rozmístění konektorů na desce plošných spojů.....	50
Obr. 43: Popis konektorů pro připojení snímaných perifériích .....	51
Obr. 44: Konektor RESET, konektory s napájecím napětím a konektory pro komunikaci s osobním počítačem .....	51

Obr. 45: Konektor rozhraní JTAG pro obvod FPGA .....	52
--	----

# SEZNAM TABULEK

Tab. 1: Možnosti konfigurace obvodu SN74ALVC164245, převzato z [10].....	6
Tab. 2: Základní vlastnosti obvodu Spartan-3E XC3S100E, převzato z [16].....	11
Tab. 3: Základní parametry mikrokontroléru ATmega32, převzato z [3] .....	13
Tab. 4: Základní vlastnosti vnitřního A/D převodníku, převzato z [3] .....	14
Tab. 5: Parametry standardů I2C, převzato z [6].....	24
Tab. 6: Význam jednotlivých pinů konektoru Cannon 9, převzato z [20] .....	27
Tab. 7: Minimální a maximální možný dosažený nasnímaný čas pro různé vzorkovací frekvence .....	31
Tab. 8: Ukázka uloženého souboru .....	36
Tab. 9: Seznam použitelných klávesových zkratk v aplikaci analyzátoru.....	43
Tab. 10: Význam prvního řídícího bytu .....	44
Tab. 11: Význam druhého řídícího bytu .....	44
Tab. 12: Význam třetího řídícího bytu .....	45
Tab. 13: Význam čtvrtého řídícího bytu .....	45
Tab. 14: Význam pátého řídícího bytu .....	45
Tab. 15: Shrnutí parametrů vytvořeného analyzátoru .....	53
Tab. 16: Shrnutí parametrů porovnávaného analyzátoru AXIS SIGMA2, převzato z [25] .....	53
Tab. 17: Shrnutí parametrů porovnávaného analyzátoru SALEAE, převzato z [26] .....	54

# ÚVOD

Cílem této diplomové práce je navrhnout a realizovat univerzální logický analyzátor vytvořený s použitím obvodu FPGA. Analyzátor umožňuje zobrazení naměřených hodnot na integrovaném displeji, nebo na osobním počítači. Před zadáním této práce byla potřeba analyzovat datové sběrnice s vysokou rychlostí a získané výsledky graficky interpretovat. To byly hlavní impulsy, které vedly k vypracování tohoto projektu. Hlavním požadavkem byla vysoká rychlost snímání, proto je jako hlavní akviziční člen zvolen obvod FPGA. Dalším důležitým požadavkem bylo propojení vytvořené aplikace s osobním počítačem. To bylo vyřešeno pomocí standardu USB díky jeho masovému rozšíření a snadné instalaci ze strany uživatele. K interpretaci naměřených údajů na osobním počítači byla využita vlastní vytvořená aplikace ve vývojovém prostředí C++Builder. Tímto programem je umožněno sledování komunikace po datových rámcích, dekódování některých základních sběrnic, uložení naměřených údajů, zobrazení nejen logické úrovně, ale i napěťové úrovně signálu, možnost zachycení pouze zvolenou strukturu rámce a možnost řízení logického analyzátoru.

Práce je členěna do osmi částí. V první části lze nalézt popis logického analyzátoru a bližší rozbor jeho možných realizací. Druhá část se zabývá přímo popisem jednotlivých funkčních bloků vybrané realizace. Konkrétně se v ní lze dočíst o návrhu napájecí části, popisu jednotlivých modulů pro změnu logických úrovní, popisu obvodu FPGA, popisu mikrokontroléru ATmega32, popisu zapojení LCD displeje a popisu dotykové fólie. Třetí část je věnována jednotlivým standardům, které je schopen analyzátor snímat jako RS232, RS485, SPI a I<sup>2</sup>C. V následující části je popsána krátká historie obvodů FPGA, možnosti jejich programování a také je zde důkladně popsán vytvořený subsystém v obvodu FPGA. Pátá část je zaměřena na vytvořenou počítačovou aplikaci. Lze se z ní dozvědět o všech funkcích vytvořené aplikace, o možnostech ovládání a také jsou zde umístěné praktické ukázky přímo z vytvořené aplikace. Mezi vytvořené funkce, které jsou zde popsány, patří např. volba snímacího režimu, použití kurzorů, ovládání interpretovaných výsledků, ukládání a nahrávání nasnímaných hodnot, vytvoření zdrojového kódu pro mikrokontrolér ATmega32 podle nasnímaného signálu, tisk a použití voltmetru. V šesté části je popsán vytvořený algoritmus pro mikrokontrolér ATmega32. Sedmá část slouží jako popis pro připojení jednotlivých rozhraní k analyzátoru. V osmé části jsou porovnány dosažené parametry s komerčně dostupnými logickými analyzátory.

# 1 Popis a realizace analyzátoru

## 1.1 Popis logického analyzátoru

Logický analyzátor je elektronické zařízení, které se používá pro identifikaci logických stavů na datových sběrnících nebo na samostatných datových vodičích. Připojením k měřenému obvodu analyzátor detekuje dva stavy: vysokou a nízkou logickou úroveň. Hlavním důvodem použití takového analyzátoru je jeho schopnost uložení několika předešlých stavů a tím názorné zobrazení datové komunikace po datových rámcích, nebo po jednotlivých bitech. Tím lze přesně analyzovat probíhající datovou komunikaci. Další možností analyzátoru je sledování pouze zadané sekvence bitů. Tím lze efektivně zobrazit, jestli ve zkoumané komunikaci jsou data, která jsou pro nás důležitá. Zobrazení datových stavů probíhá na počítači nebo na displeji přístroje. Díky těmto vlastnostem nachází toto zařízení velké spektrum uplatnění a je nezbytné pro rozsáhlé analýzy datové komunikace. Univerzálním logickým analyzátozem lze sledovat několik odlišných rozhraní jako např. I<sup>2</sup>C, SPI, RS232 apod.

Na logické analyzátozy jsou kladeny různé požadavky, které se liší podle druhu použití a konstrukce. Mohou být přenosné, stolní, napájené bateriemi, napájené ze sítě apod. Důležitým parametrem je maximální rychlost záznamu, velikost paměti a počet vstupních kanálů. Podle těchto požadavků většinou volíme způsob realizace. Pro zachycení datové komunikace o rychlosti v řádu stovek kilohertzů až jednotek megahertzů lze použít mikrokontrolér. Pro rychlejší datové komunikace se již využívají například obvody typu PLD, mikrokontroléry typu DSP, FPGA – (Field Programmable Gate Arrays), nebo k tomuto účelu přímo uzpůsobené obvody ASIC.

K vytvoření logického analyzátoru lze použít několik možných realizací. Za zmínku stojí především realizace pomocí klasických logických hradel a sekvenčních obvodů, realizace pomocí mikrokontrolérů a jako poslední realizace pomocí obvodů FPGA. První zmíněná možnost realizace je zatížena mnoha nevýhodami, proto se dnes již vůbec nevyužívá, díky tomu následující srovnání obsahuje pouze porovnání mikrokontrolérů s obvody FPGA. Při výběru je nutné porovnávat cenu realizace, rychlost vývoje, dosažitelný výkon, ochranu duševního vlastnictví a velikost výrobní série.

## 1.2 Možnosti realizace

Jako první lze využít některý z komerčně dostupných mikrokontrolérů. Jedná se o poměrně levnou variantu, která se spíše používá pro menší výrobní série. Programování probíhá buď ve vyšších programovacích jazycích, anebo přímo v assembleru. Rychlost vývoje je díky lehkému programování a možnosti přenesení programu ve vyšším programovacím jazyku i na jiné mikrokontroléry poměrně rychlá. Omezením této varianty je především výpočetní výkon. Ten se udává v MIPS – (Millions of Instruction Per Second), to znamená milióny instrukcí za sekundu. Maximální výkon mikrokontroléru je udán maximální možnou hodinovou frekvencí a počtem funkčních jednotek. Funkční jednotkou se myslí např. ALU – (Arithmetic Logic Unit), nebo MAC – (Multiply-and-ACcumulate). Přidáváním dalších jednotek nelze neustále zvyšovat výkon, protože s každou další jednotkou se zvyšují kapacitní zátěže sběrnic, a tím je nutné snížit maximální hodinový kmitočet. Dále je potřeba, aby kompilátor dokázal optimalizovat výkon na všechny jednotky, protože je velmi obtížné vytvořit vlastními silami algoritmus, který by tuto optimalizaci dokázal. Dalším problémem je, že některé jednotky

nejsou k aplikaci potřeba, a proto jsou zcela nevyužité. Poslední nevýhodou tohoto návrhu je nízká ochrana duševního vlastnictví. V podstatě stačí okopírovat použitý program.

Druhou zmíněnou variantou je vytvoření aplikace pomocí obvodu FPGA. Pro použití ve větších sériích je tento obvod poměrně drahý, ale pro menší série je vhodnější než obvody ASIC. V porovnání s mikrokontroléry jsou obvody FPGA finančně náročnější. Jejich výkon se udává v MMACs – (Millions of Multiply-and-ACCumulate operations per Second), což znamená miliony MAC operací za sekundu. Někdy bývá tato hodnota výrobcem nadsazena, protože se k ní často dospěje jako počet násobiček na FPGA vynásobených maximálním hodinovým kmitočtem násobičky. Hodnotu v MMACs má smysl uvádět pouze u aplikací DSP. Pro naši realizaci je tato hodnota neúčinná. I přes nadsazená čísla jsou reálné hodnoty výkonů díky vysoké paralelizaci o řád vyšší než u nejlepších signálových procesorů označovaných jako DSP – (Digital Signal Processor). Rychlost vývoje aplikace je pomalejší než u mikrokontroléru a také obtížnost návrhu je podstatně vyšší. Při návrhu je nutné pamatovat, že i při použití vyšších programovacích jazyků se jedná pořád o návrh hardwaru. [1]

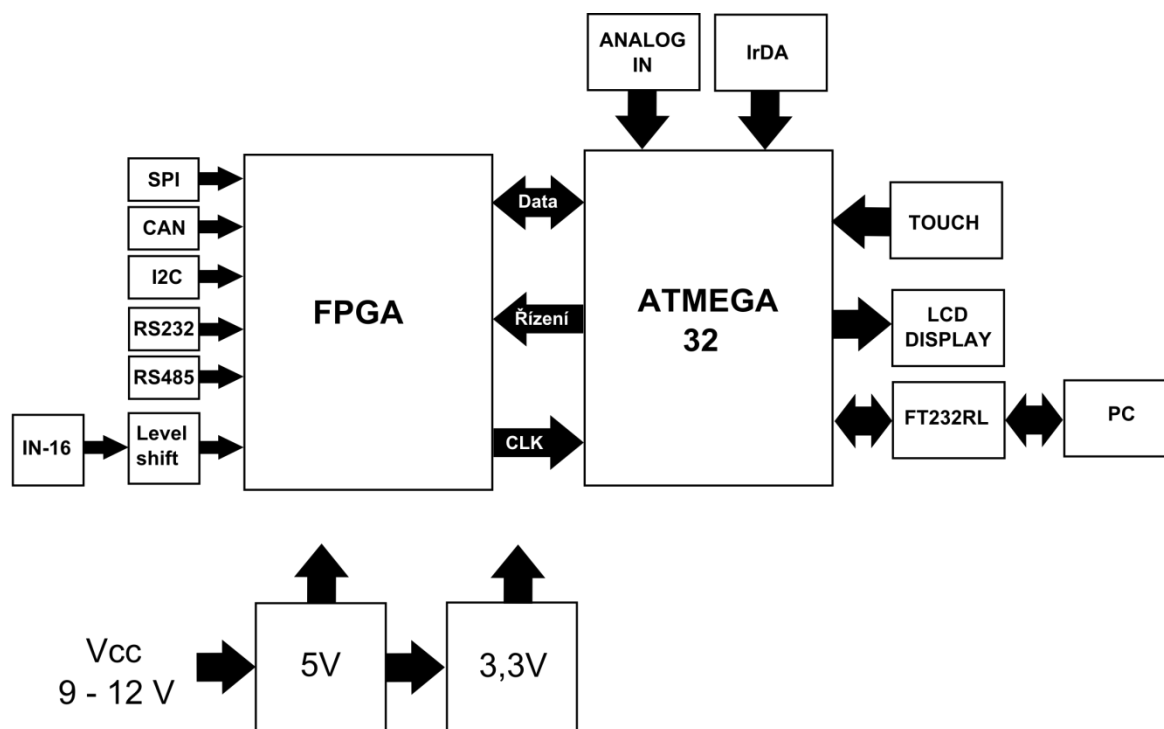
Výběr realizace tohoto logického analyzátoru je ovlivněn především těmito požadavky: nízká cena při kusové výrobě, vysoká rychlost snímání, možnost připojení osobního počítače, i displeje s dotykovou fólií, poměrně rychlá realizace a také použití dostupných výrobních možností.

Použitím pouze mikrokontroléru by i přes rychlý vývoj těchto obvodů nebyla dodržena podmínka vysoké vzorkovací frekvence. Jako příklad lze uvést 8bitový mikrokontrolér řady AVR ATmega32 a zástupce řady ARM STM32F4. První zmiňovaný mikrokontrolér ATmega32 pracuje s maximálním hodinovým kmitočtem 16 MHz. Pokud vezme v úvahu, že získání logických stavů na jediném portu obvodu a následné uložení trvá minimálně dvacet hodinových taktů a minimální vzorkovací frekvence musí být alespoň dvakrát vyšší, dostaneme snímací rychlost kolem 400 kHz. Tato rychlost je pouze pro jeden 8bitový port. Pokud bychom chtěli snímat více sběrnic, tak je nutné použít více portů, a tím by se také úměrně snižovala vzorkovací frekvence. Mikrokontrolér STM32F4 může pracovat s hodinovým kmitočtem až 168 MHz, tím lze dosáhnout zvýšení vzorkovací frekvence až k hranici čtyř megahertzů. Tato hodnota je také pro naše použití v logickém analyzátoru nízká.

Požadavek na vzorkovací frekvenci v řádu desítek megahertzů splňuje pouze realizace s použitím obvodu FPGA. Ovšem zde není splněna podmínka jednoduchého připojení periférií jako displej, nebo osobní počítač pomocí USB. Díky těmto vlastnostem je navržený logický analyzátor složen z obou zmiňovaných obvodů. Na snímání logických stavů sběrnic je využit obvod FPGA, který předává dál naměřená data mikrokontroléru ATmega32. Tento mikrokontrolér údaje zobrazí na displeji, nebo pokud je připojen osobní počítač, tak ve vytvořené aplikaci. Další výhodou použití mikrokontroléru je jednoduché připojení dotykové fólie nutné k ovládání přímo na vnitřní A/D převodník. Z důvodů složitosti návrhu a výroby desky plošných spojů pro obvod FPGA je logický analyzátor tvořen již zakoupenou univerzální deskou, která tento obvod obsahuje. Kombinací levnějšího obvodu FPGA a také méně finančně náročného mikrokontroléru se s cenou dostaneme k rozumným hodnotám a jsou splněny i nutné požadavky pro toto zařízení.

## 2 Popis jednotlivých bloků a jejich realizace

Univerzální logický analyzátor se skládá z několika funkčních bloků. Některé bloky jsou tvořené pouze softwarově jako například blok I<sup>2</sup>C, SPI anebo pouze s minimem okolních součástek jako IrDA a také analogové vstupy. Ostatní funkční bloky jsou vždy hardwarovým řešením. Pro jednotlivé logické standardy obsahují bloky především překladače logických úrovní. Dále jsou tu funkční bloky jako Atmega32, FPGA, LCD displej, dotyková fólie a obvod pro konverzi na USB FT232RL. Další nezbytnou částí zařízení je napájení, které je tvořeno dvěma obvody, z kterých je odebíráno napětí o nominální hodnotě 5 V a 3,3 V. V následujících kapitolách jsou podrobně rozepsány jednotlivé funkční bloky. Blokové schéma obvodu lze vidět na obr. 1.



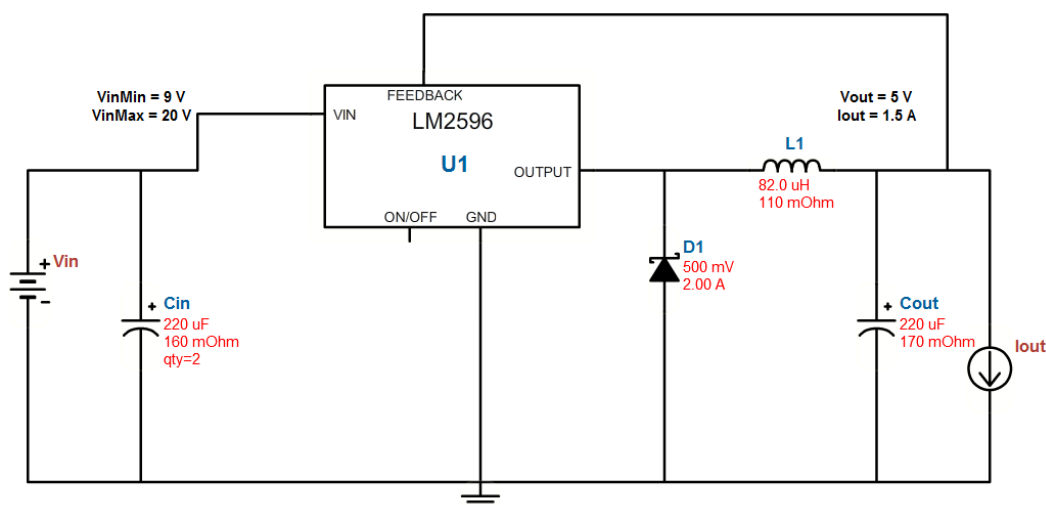
Obr. 1: Blokové schéma navrhnutého obvodu

Jako první částí analyzátoru jsou bloky, které převádí logické signály z daného standardu na logickou úroveň 3,3 V, která je nutná pro programovatelný logický obvod typu FPGA. U bloků RS232 a RS485 je toho docíleno přímo vybraným obvodem, přičemž u bloku CAN je ještě nutné přidat k obvodu tranzistory pro změnu logické úrovně. Konverzi logických úrovní pro blok GPIO zajišťuje blok LEVEL SHIFT, který obsahuje měnič napěťové úrovně. Převezené signály poté pokračují do bloku FPGA, kde jsou zpracovány a dále poslány do bloku ATMEGA32. Tento blok zpracovává signály, které přicházejí jak z bloku FPGA, tak také z bloků ANALOG IN, IrDA, PC a TOUCH, což je dotyková fólie nutná pro ovládání zařízení pomocí displeje. Nasnímané signály blok ATMEGA32 zpracuje a dále distribuuje do bloku LCD DISPLAY, nebo do PC přes blok FT232RL konvertující USART do formátu USB. V PC jsou opět naměřená data zpracovány pomocí aplikace, která byla vytvořena v C++ Builderu. Komunikace mezi bloky PC a ATMEGA32 je obousměrná pro možnost ovládat

zařízení přímo pomocí osobního počítače. Poslední bloky distribuují napájecí napětí o hodnotách 5 V a 3,3 V.

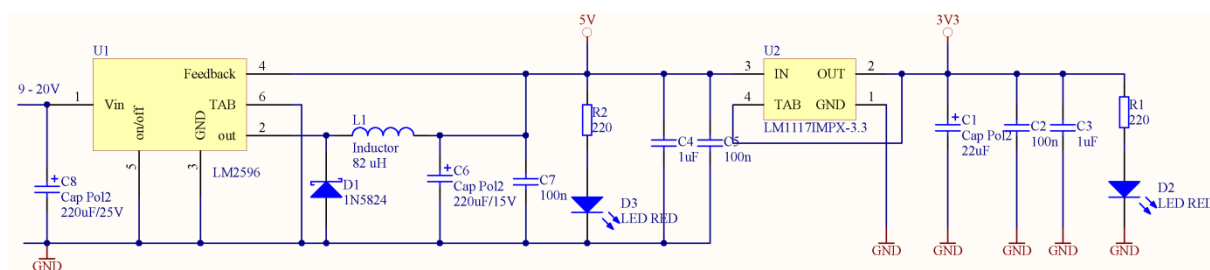
## 2.1 Napájecí část

Napájecí napětí se může pohybovat v rozmezí 9 – 20 V. Toto napětí je jako první přivedeno na pěti voltový spínaný měnič LM2596 a poté na lineární měnič LM1117, který má výstupní napětí o hodnotě 3,3 V. Navržené hodnoty součástek, které jsou nutné pro funkci spínaného měniče LM2596 jsou odečteny z návrhového systému od firmy Texas Instruments. Optimalizace z návrhového systému lze vidět na obr. 2.



Obr. 2: Optimalizace navrhnutého spínaného zdroje, převzato z [12]

Napájecí blok, kterým jsou napájené všechny obvodové části je vidět na následujícím obrázku obr. 3.



Obr. 3: Napájecí část obvodu

## 2.2 IN-16

V bloku IN-16 je zahrnuta 15bitová datová sběrnice a jeden pin určený pro hodinový signál. Tento počet pinů je omezen především šestnácti pinovým měničem napěťové úrovně, který přímo navazuje na tento blok. Tato sběrnice lze použít jako univerzální synchronní a asynchronní datová sběrnice. Další použití této sběrnice je pro snímání standardů I<sup>2</sup>C, SPI a



UART. Pro připojení hodinového signálu standardu I<sup>2</sup>C je použit pin IN2 a pro datový signál IN3. Pro připojení SPI, slouží piny: IN4 pro hodinový signál, IN5 pro signál MISO, IN6 pro signál MOSI a IN7 pro signál CS. K snímání standardu UART je použit pin IN8.

## 2.3 Měnič napěťové úrovně SN74ALVC164245

V zařízení je použit obousměrný měnič úrovně od firmy Texas Instruments SN74ALVC164245. Jeho aplikace je vhodná tam, kde je nutné snížit, nebo naopak zvýšit napětí z 5 V na 3,3 V. Obsahuje šestnáct pinů pro každé napájecí napětí. O řízení tohoto obvodu se starají dva piny obvodu a to: DIR, který řídí směr a OE, který povoluje výstup obvodu. Pro názornost je princip řízení zobrazen v následující tabulce.

Tab. 1: Možnosti konfigurace obvodu SN74ALVC164245, převzato z [10]

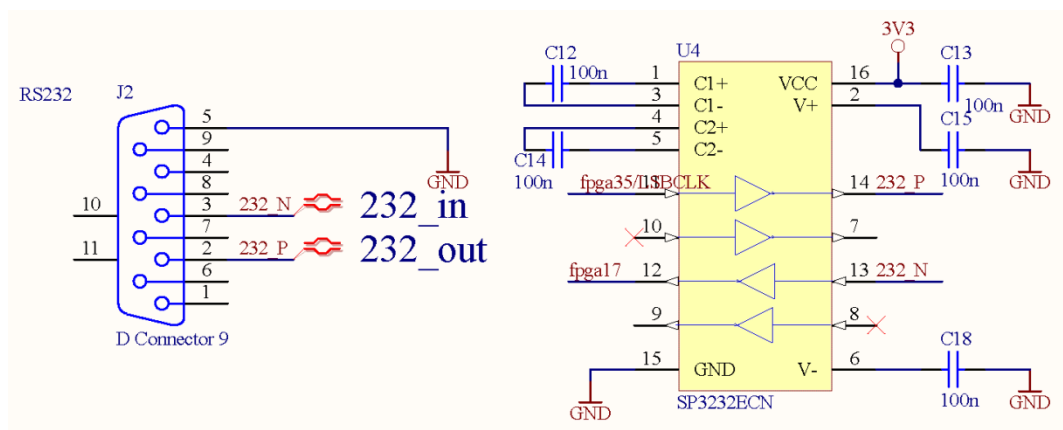
Řídící vstup		Výstup obvodu		Funkce
OE	DIR	A Port	B Port	
L	L	Povolen	Vysoká Impedance	Směr: z B do A
L	H	Vysoká Impedance	Povolen	Směr: z A do B
H	X	Vysoká Impedance	Vysoká Impedance	Izolace

Díky tomu, že je tento obvod jako první blok na vstupní straně, je na něm provedeno ošetření proti přepětí a proti záporným impulsům, které by mohli poškodit obvod. Případné vyšší napětí omezuje transil SP0502BAJTG a záporné impulzy omezuje schottkyho dioda BAS70.

## 2.4 Fyzické rozhraní RS232, RS485 a CAN

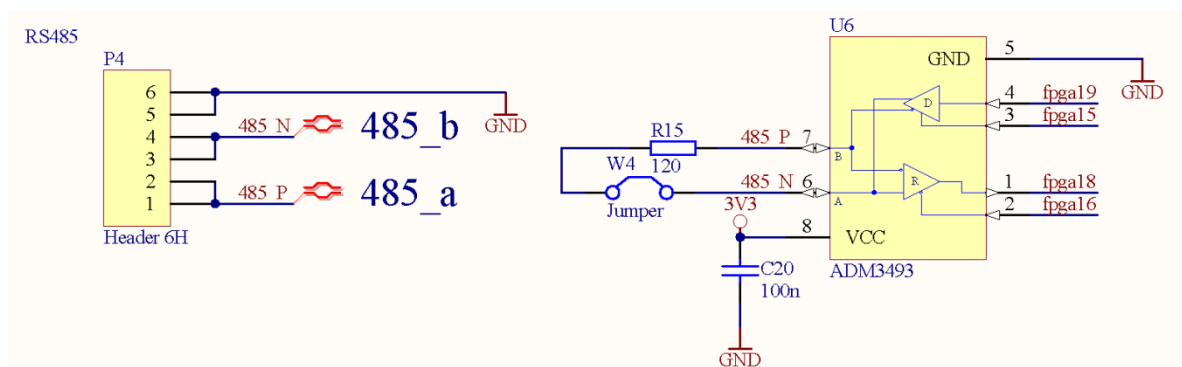
Jelikož je tento logický analyzátor také určen přímo pro dekódování konkrétních datových standardů, je potřeba upravit jednotlivé logické úrovně vždy na napěťové úrovně 0 až 3,3 V, které dokáže zpracovat obvod FPGA. Proto je nutné ke každému snímanému standardu předřadit fyzické rozhraní, které dokáže tuto konverzi provést. Tyto rozhraní většinou obsahují pouze integrovaný obvod s minimem okolních součástek.

Jako první je nutné upravit logické úrovně standardu RS232. K tomuto účelu je použit obvod SP3232ECN. Zapojení okolních součástek a připojení k obvodu FPGA lze nalézt na obr. 4.



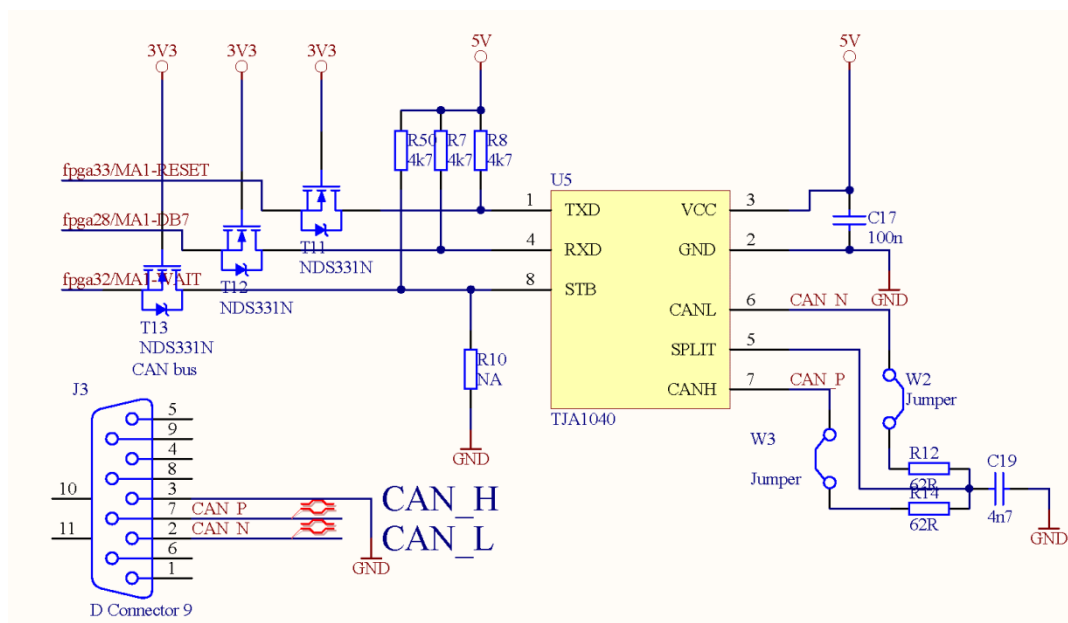
Obr. 4: Zapojení obvodu SP3232ECN, převzato z [12]

Dále je nutné převést standard RS485 a to pomocí obvodu ADM3493. Zapojení tohoto převodníku ukazuje Obr. 5.



Obr. 5: Zapojení obvodu ADM3493, převzato z [13]

Jako poslední je v zapojení použit integrovaný obvod TJA1040 pro konverzi sběrnice CAN. Obvodové schéma je zapojeno podle obr. 6. Tranzistory T11 až T13 jsou zde použity pro změnu úrovně z 5 V na 3,3 V. Součástky s hodnotou NA nejsou v zařízení osazeny, ale jsou zde pouze pro případ možnosti použití.

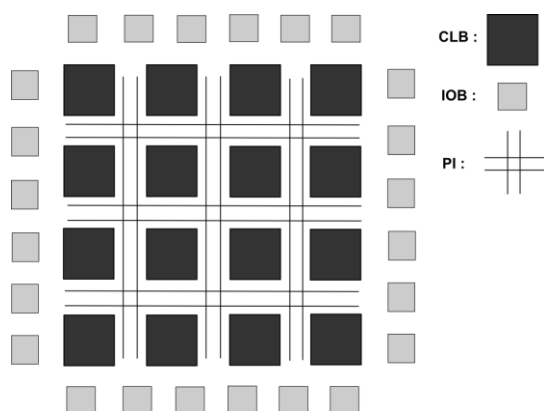


Obr. 6: Zapojení obvodu TJA1040, převzato z [14]

## 2.5 Obvod FPGA

Pod pojmem FPGA se skrývá obvod známý jako programovatelné hradlové pole. Díky přednostem těchto obvodů jsou v dnešní době stále více aplikovány do různých zařízení. Mezi jejich hlavní přednosti patří především: univerzálnost, rychlost přeprogramování, paralelismus, přechod na zákaznické obvody typu ASIC a mnoho možností různých typů návrhu daného systému. Již zmíněná paralelizace zpracování umožní o řád vyšší rychlost zpracování informace oproti nejlepším mikrokontrolérům typu DSP. Tím je lze využít i při zpracování větších objemů dat např. multimediální signály apod. Dále jsou tyto obvody dosti užívány pro realizaci algoritmů, které mají později sloužit jako podklad pro zákaznické obvody ASIC, nebo jako jejich náhrada při malých výrobních sériích. Také je lze využít, pokud jde o rychlost uvedení zařízení na trh a vývoj ASIC obvodu by byl časově náročný. [1]

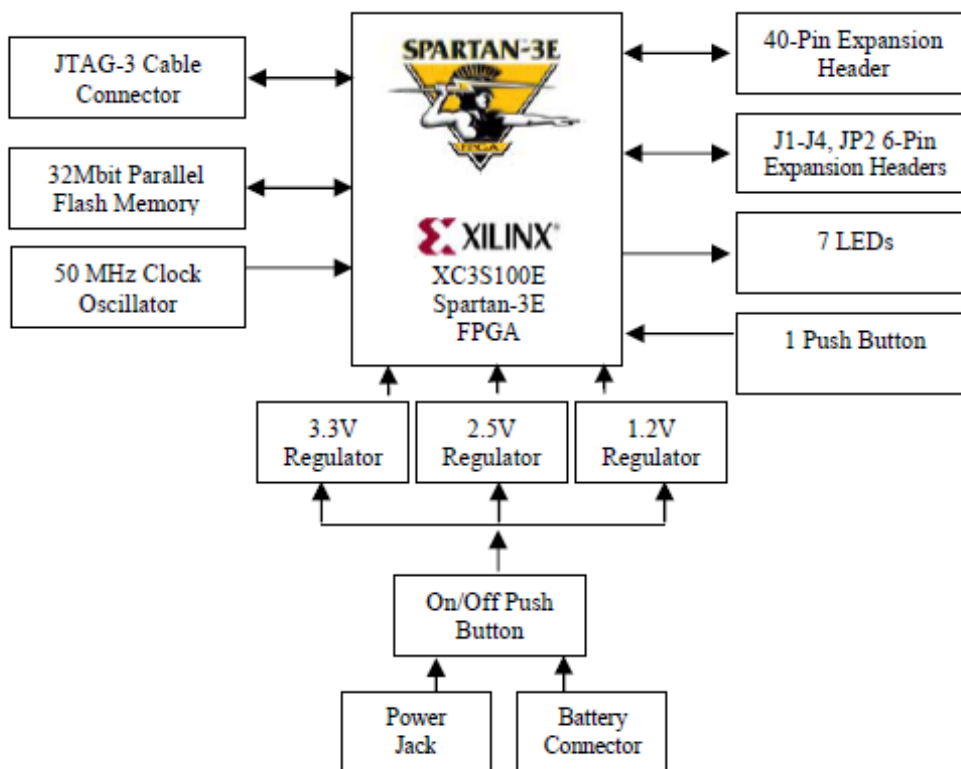
FPGA obvody jsou představitelé skupiny obvodů s tzv. „jemnozrnnou“ strukturou. Tato struktura se vyznačuje tím, že obsahuje malé základní funkční bloky, které se navzájem propojují. Zjednodušené vnitřní zapojení obvodu FPGA znázorňuje obr. 7.



Obr. 7: Vnitřní zapojení obvodu FPGA, převzato z [2]

Z obrázku je patrné, že je obvod dělen na tři části. Jako základ je použito pole konfigurovatelných logických bloků CLB – (Configurable Logic Block). Bloky jsou tvořeny na principu paměti PROM a nazývány LUT – (Look-Up Table), kde lze vytvořit všechny kombinační funkce pro vstupní signály. Obvykle obsahují čtyři vstupy, což znamená vytvoření 16bitové paměti. V dnešní době jsou již LUT i se šesti vstupy. Pokud je potřeba větších logických funkcí, tak je nutné připojit k jedné LUT další blok s LUT. CLB nemusí obsahovat pouze LUT, ale také přímo specializované obvody jako např. multiplexory. Dále vedle bloků CLB obvody FPGA obsahují také paměti RAM/ROM, různé struktury pro sériovou komunikaci apod. Veškeré bloky propojuje struktura PI – (Programmable Interconnect). Díky rozsáhlosti propojovací struktury je poměr zabraného místa na čipu mezi CLB a PI jeden ze základních faktorů, který určuje efektivnost obvodu. Posledním prvkem v obvodu jsou prvky IOB – (Input/Output Block). Tyto bloky vytváří možnost propojení obvodu s dalšími obvody či periferiemi. Obsahují obvykle pouze klopné obvody. Zobrazená struktura na obr. 7 ovšem platí pouze pro obvody, které mají vývody po svém obvodu. Pokud je tento obvod vyrobený jinou technologií a obsahuje kontaktní piny různě rozmístěné pod čipem, mohou mít výstupní bloky IOB různé rozmístění. [2]

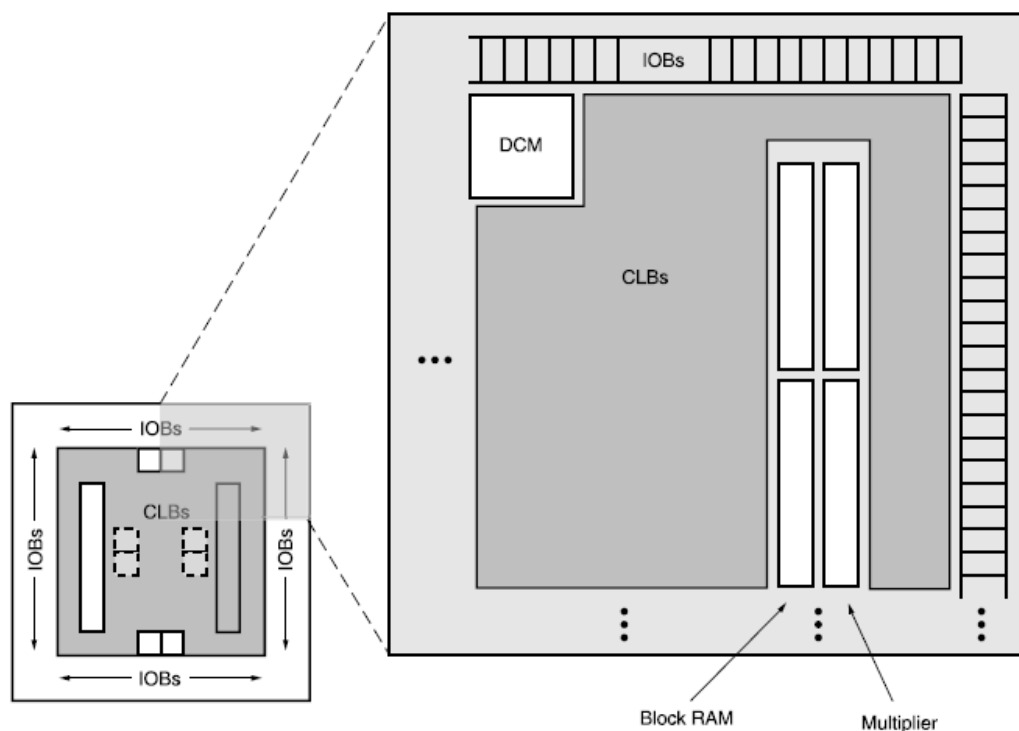
V zařízení je použitý vývojový kit, který je osazen obvodem FPGA Spartan-3E XC3S100E od firmy Xilinx. Tento vývojový kit obsahuje sedmdesát rozšiřujících pinů, sedm LED diod, konektor pro připojení JTAG rozhraní, tlačítko pro univerzální použití, tlačítko pro ovládání spouštění kitu, 50 MHz oscilátor, 32 Mbit Flash paměť a obvody pro úpravu napájecího napětí. Všechny tyto bloky lze vidět také na obr. 8. [16]



Obr. 8: Blokové zapojení vývojového kitu s obvodem FPGA, převzato z [15]

Obvody rodiny Spartan-3E obsahují pět základních programovatelných funkčních bloků a to: konfigurovatelné logické bloky (Configurable Logic Block, CLB), vstupně-výstupní bloky (Input/Output Block, IOBs), blok RAM, multiplikační bloky a digitální hodinový manažer (Digital Clock Manager, DCM).

Konfigurovatelné logické buňky (CLBs) obsahují programovatelné tabulky LUT (Look-Up Tables), které realizují kombinační funkce. Dále obsahují klopné obvody, které lze nakonfigurovat jako řízené hranou (Flip-Flop, FF), nebo staticky řízené (Latch). Buňky umožňují vytvoření logických funkcí a jsou dobré také pro ukládání dat. Vstupně-výstupní bloky (IOBs) upravují úroveň signálů mezi piny a vnitřní logikou zařízení. Každý vstupně-výstupní blok obsahuje také možnost použití jako třístavového výstupu. Vstupně-výstupním pinům lze přiřadit několik napěťových standardů jako např. 3.3 V low-voltage TTL (LVTTTL), low-voltage CMOS (LVCMOS) pro úrovně 3,3 V, 2,5 V, 1,8 V, 1,5 V, nebo 1,2 V anebo 3 V PCI pro 33 MHz. Některé piny obvodu ovšem povolují použití pouze jako vstupní bránu. Bloky RAM poskytují datovou paměť, která se skládá ze dvou 18kbit částí. Bloky hardwarových násobiček podporují operace násobení se vstupními daty o délce 18 bit. Blok DCM umožňuje syntetizovat hodinové signály různých kmitočtů, dále zajišťuje distribuci hodinového signálu, umožňuje fázový posun hodinových signálů apod. Vnitřní strukturu obvodů Spartan-3E lze nalézt na obr. 9. [16]



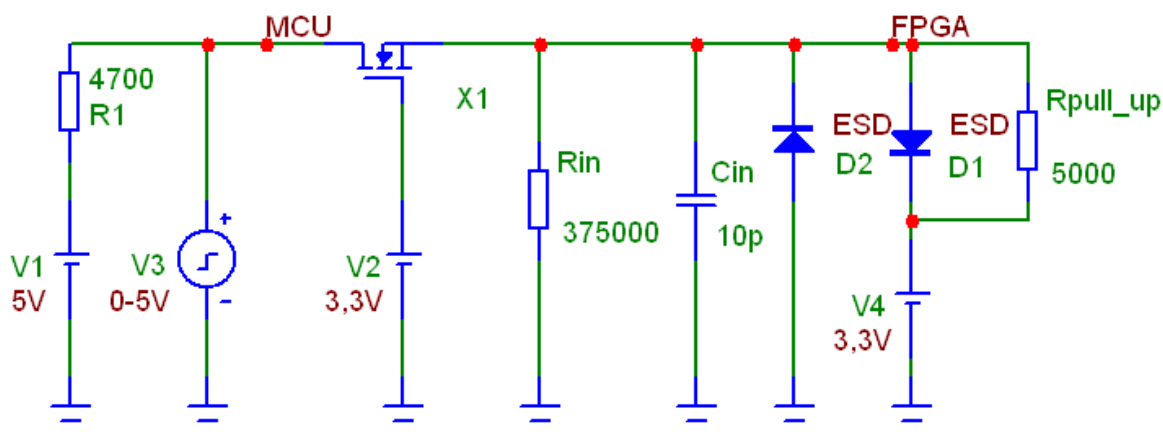
Obr. 9: Rozložení bloků v obvodu Spartan-3E, převzato z [16]

Základní vlastnosti obvodu Spartan-3E XC3S100E ukazuje následující tabulka tab. 2. [16]

Tab. 2: Základní vlastnosti obvodu Spartan-3E XC3S100E, převzato z [16]

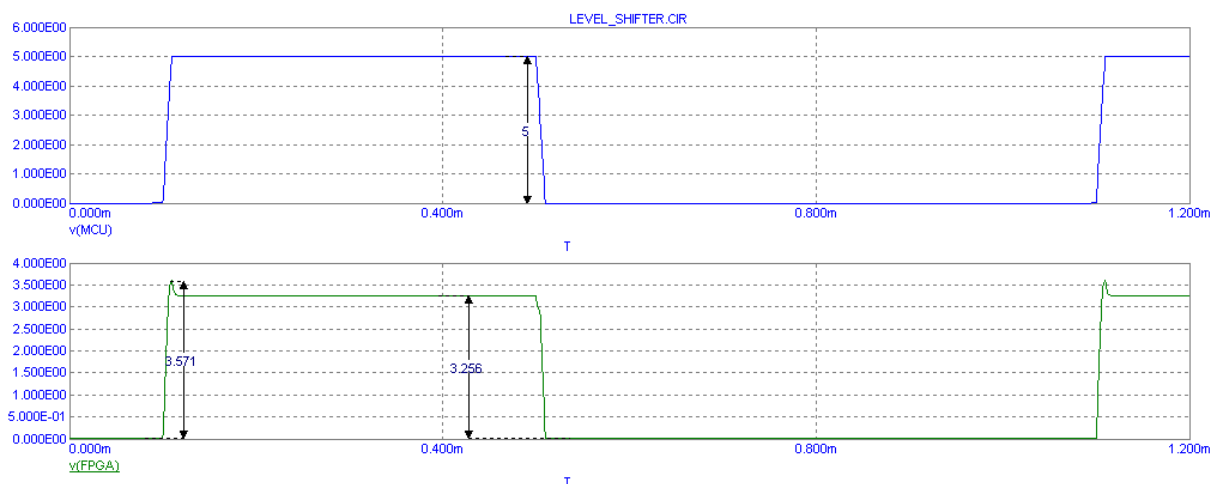
Vlastnost		Počet
Systémových buněk		100k
Ekvivalentních logických buněk		2160
Konfigurovatelných logických bloků (jeden blok = čtyři řezy)	Řádky	22
	Sloupce	16
	Celkem buněk	240
	Celkem řezů	960
Počet bitů distribuovaných v RAM		15k
Počet bitů v bloku RAM		72k
Specializovaných násobiček		4
DCM bloků		2
Celkový počet I/O pinů		108
Celkový počet diferenciálních párů		40

Z důvodů rozdílných napěťových úrovní pro vstupně-výstupní piny obvodu FPGA a mikrokontroléru ATmega32 bylo nutné do společných signálů vložit obvod pro konverzi signálů ze standardu TTL na logické signály pracující s napětím 3,3 V. Toho je docíleno pomocí MOSFET tranzistoru NDS331. Zapojení s tímto tranzistorem, se vstupním odporem, vstupní kapacitou, ochrannými diodami a pull-up rezistorem bylo odsimulováno v programu Micro-Cap. Schéma zapojení simulace lze vidět na obr. 10.



Obr. 10: Simulace obvodu pro konverzi z 5 V na 3,3 V

Na generátoru napětí V3 byl nastaven kmitočet 1 kHz a byla spuštěna časová simulace. Odsimulovaný průběh napětí na obvodu FPGA ukazuje obrázek obr. 11.



Obr. 11: Časová simulace obvodu pro konverzi z 5 V na 3,3 V

## 2.6 Mikrokontrolér ATmega32

Mikrokontroléry jsou obvody obsahující mikroprocesor s pamětí, různé rozšiřující bloky a periferie pro připojení externích součástek. Díky možnosti přeprogramování funkce a výkonu těchto obvodů jsou dnes používány v mnoha aplikacích a tvoří nemalou část dnešních elektronických zařízení.

ATmega32 patří do skupiny nízkopříkonových 8bitových mikrokontrolérů architektury AVR RISC vyráběné firmou Atmel.

Z vlastností vybraného mikrokontroléru jsou použity funkce jako komunikace USART pro připojení počítače, A/D převodník pro převod hodnot z dotykové fólie a zjištění dvou analogových napětí, vstupně-výstupní porty obvodu pro připojení displeje, komunikaci s obvodem FPGA a pro připojení dotykové fólie. V následující tabulce tab. 3 jsou shrnuty základní parametry tohoto obvodu. Jednotlivé funkce jako použití vstupně-výstupních portů, A/D převodníku a jednotky USART jsou detailně rozebrány v následujících odstavcích.

Tab. 3: Základní parametry mikrokontroléru ATmega32, převzato z [3]

<b>Obsah:</b>	<b>Popis:</b>
<b>Počet instrukcí v instrukčním souboru:</b>	131
<b>Počet osmi bitových registrů:</b>	32
<b>Počet vstupně-výstupních osmi bitových portů:</b>	4
<b>Hodinový kmitočet:</b>	0 – 16 MHz
<b>Velikost FLASH paměti pro zápis programu:</b>	32 KB
<b>Zaručený počet přeprogramování:</b>	1000
<b>Velikost datové paměti RAM:</b>	1 KB
<b>Velikost datové paměti EEPROM:</b>	512 B
<b>Zaručený počet přepsání:</b>	100 000
<b>Programování FLASH a EEPROM pomocí:</b>	SPI, JTAG
<b>Počet čítačů/časovačů:</b>	3
<b>Počet PWM kanálů:</b>	4
<b>Počet analogových komparátorů:</b>	1
<b>Počet deseti bitových A/D převodníků:</b>	1
<b>Podporované sběrnice:</b>	USART, SPI, I2C
<b>Možnosti resetu:</b>	WDT, Power-on reset
<b>Zabudovaný vnitřní RC oscilátor:</b>	1 – 8 MHz
<b>Pouzdra:</b>	DIP 40, TQFP 44

Jak je patrné z tabulky tab. 3, tak tento mikrokontrolér obsahuje čtyři 8bitové vstupně-výstupní porty. Každému vývodu v jednotlivých portech lze nastavit směr bez ovlivnění sousedních vývodů. Směr každého portu určuje registr DDRx, kde za x se dosadí označení portu od A do D. Nastavením DDRx do stavu log. 0 určíme, že nastavovaný port je brán jako vstupní. Naopak při stavu log. 1 je nastaven port jako výstupní. Pokud máme port nastavený jako výstupní, tak zápis dat probíhá pomocí registru PORTx. Registrem PINx lze naopak číst data z námi vybraného portu. Posledním zmiňovaným bitem spojeným s vstupně-výstupními porty je bit PUD z registru SFIOR. Tímto bitem lze nastavit odpojení pull-up rezistorů pro všechny vstupně-výstupní porty. Pokud nastavíme na tento bit stav log. 1 budou odpojeny všechny pull-up rezistory. V opačném případě tento bit nijak neovlivní funkci pull-up rezistorů. V případě že používáme port jako vstupní, tzn. DDRx je ve stavu log. 0, tak nastavení pull-up rezistoru zajišťuje registr PORTx, kde platí, že pokud je tento registr nastaven ve stavu log. 1, tak jsou pull-up rezistory připojené. U tohoto mikrokontroléru není nutné řešit, v kterém stavu lze budit např. LED diody, protože má symetrické charakteristiky pro oba logické stavy. Maximální proud jednoho výstupu nesmí překročit hodnotu 40 mA a součet všech výstupů nesmí překročit hodnotu 200 mA. Dále také kromě použití portu jako vstupně-výstupního zařízení lze tento port použít pro sekundární funkci, která je vždy



přiřazena konkrétnímu vývodu. Může se jednat třeba o vstup A/D převodníku, nebo vývody nutné pro USART komunikaci apod. [3]

Jako další je použita v aplikaci funkce vnitřního A/D převodníku, která slouží jako dekodér dotykového ovládání a k měření dvou analogových napětí. Některé základní parametry použitého A/D převodníku jsou v tab. 4.

Tab. 4: Základní vlastnosti vnitřního A/D převodníku, převzato z [3]

<b>Funkce:</b>	<b>Popis</b>
<b>Bitové rozlišení:</b>	10
<b>Integrální nelinearita:</b>	0,5 LSB
<b>Absolutní chyba:</b>	$\pm 2$ LSB
<b>Doba převodu:</b>	65 – 260 $\mu$ s
<b>Rychlost převodu:</b>	Max. 15 kSPS
<b>Multiplexovaných vstupních kanálů:</b>	8
<b>Diferenčních vstupní kanálů:</b>	7
<b>Volitelná zabudovaná reference:</b>	2,56V

Vnitřní A/D převodník je 10bitový a je vybaven analogovým multiplexorem, který přepíná až deset různých vstupů, které lze měřit i diferenčně. Dále jsou hodnoty vzorkovány vzorkovačem S/H – (Sample and Hold) a jsou připojeny na zesilovač, který drží po celou dobu stálou úroveň. Napájení převodníku zajišťují samostatné vývody. Pro napájecí napětí slouží vývod  $AU_{CC}$  a pro připojení země je určen vývod AGND. AGND je analogová zem, která musí být připojena s GND pouze v jediném bodě. Připojené napětí na vývodu  $AU_{CC}$  musí být v rozsahu  $U_{CC} \pm 0,3$  V. Pokud není vybráno diferenční měření mezi vývody, tak měření probíhá mezi vývodem AGND a zvolenou referencí. Jako referenční napětí může být použito napětí  $AU_{CC}$ , přivedené napětí na vývod AREF a nebo také vnitřní referenční napětí 2,56 V. Vždy je nutné na vývod AREF připojit blokovací kondenzátor i pokud vývod AREF není použit jako referenční zdroj. Použití referenčního napětí lze měnit v registru ADMUX a to bity REFS0 a REFS1. Dále registr ADMUX obsahuje registry MUX0 až MUX4, kterými volíme zisk a výběr vstupního analogového kanálu. Všechny vstupy a také i AGND a vnitřní referenci 2,56 V lze měřit proti vývodu AGND. Pokud chceme použít diferenčního měření rozdílového napětí, lze využít některé vývody, u kterých lze volit i zesílení. Pro měření proti AGND je zisk vždy roven jedné. Povolení A/D převodníku je provedeno nastavením bitu ADEN v registru ADCSRA do stavu log. 1. V opačném případě je A/D převodník vypnut. V tomto režimu A/D převodník nespotřebovává žádnou energii, tím lze dosáhnout úspory energie v režimu spánku. Po převodu jsou data uloženy do registrů ADCH a ADCL. Jako první se musí číst data z registru ADCL, aby bylo zajištěno, že hodnota ADCH patří ke stejné hodnotě jako ADCL. Hodnotu ADCL nelze číst dvakrát. Další přečtení registru ADCL je možné až po přečtení registru ADCH. Výsledná interpretovaná hodnota je implicitně zarovnaná doprava. Pro možnost opačného zarovnání je vyhrazena volba bitu ADLAR

v registru ADMUX. Při použití druhého způsobu řazení lze pro případ potřeby jen 8bitové přesnosti použít pouze registr ADCH. Po ukončení převodu analogové hodnoty na digitální je aktivováno přerušení.

Spouštěč A/D převodník lze třemi způsoby, které se dělí na režimy: jednoduchý převod, automatické spouštění a volný běh. První režim A/D převodníku je řízen bitem ADSC v registru ADCSRA. Nastavením tohoto bitu do stavu log. 1 započneme převod. Vynulování tohoto bitu se provede automaticky hardwarově. Další možností je použít automatického spouštění. Tento režim je povolený nastavením bitu ADATE do stavu log. 1 v registru ADCSRA. Výběr spouštěcího zdroje je voleno bity ADTS0 až ADTS2 v registru SFIOR. Začátek převodu započne s náběžnou hranou vybraného zdroje spouštění. Další náběžná hrana zvoleného zdroje je ignorována pokud již probíhá převod. Jako poslední možností je použití volného běhu. Spuštění prvního převodu je nutné vyvolat opět ručně a to nastavením bitu ADSC do stavu log. 1 v registru ADCSRA. Poté se po ukončení předchozího převodu automaticky aktivuje následující převod.

Pro správnou funkci A/D převodníku při použití všech 10bitů je potřeba hodinový kmitočet rozmezí 50 kHz až 200 kHz. Pro menší rozlišení je možné tento hodinový kmitočet navýšit, ale do maximální hodnoty 1 MHz. Pro získání těchto hodinových kmitočtů je do mikrokontroléru přidána předdělička kmitočtu. Výběr hodnoty kmitočtu předděličky je určena bity ADPS0 až ADPS2 v registru ADCSRA. Tento hodinový kmitočet započne s aktivací A/D převodníku, tzn. s nastavením bitu ADEN v registru ADCSRA do stavu log. 1. Nulování čítání opět řídí bit ADEN a to nastavením do stavu log. 0. První převod analogové hodnoty do digitální reprezentace trvá dvacet pět hodinových taktů a každý další následující převod pouze třináct. Při použití režimu volného běhu a při zvoleném kmitočtu 200 kHz lze vzorkovat přibližně 15 kSPs. Zvláštní pozor je nutné dát na maximální možný vstupní kmitočet diferenčního zesilovače, který je 4 kHz. Při vyšších frekvencích se projevuje nelinearita zesilovače. Je vhodné tyto vstupy ošetřit dolní propustí, která dle doporučení autora by měla mít mezní kmitočet alespoň desetkrát nižší.

Další zajímavou funkcí vnitřního A/D převodníku je možnost použití v IDLE módu mikrokontroléru. Tím je zajištěno menší rušení z jádra mikrokontroléru, které může pronikat do A/D převodníku. V tomto režimu je nutné dodržet následující postup. Jako první musí být A/D převod zapnut, musí být také nastaven režim jednoduchého převodu a musí být povoleno přerušení po dokončení převodu. V dalším kroku je mikrokontrolér převeden do režimu IDLE a začíná převod hodnoty. Po převodu se automaticky opět aktivuje mikrokontrolér z režimu IDLE.

Výsledná hodnota po dokončení převodu je v registrovém páru ADCH:ADCL. Pro její převod platí rovnice (2.1), kde  $U_{VST}$  znamená vstupní napětí a  $U_{REF}$  je referenční hodnota napětí.

$$ADCH:ADCL = 1024 \cdot \frac{U_{VST}}{U_{REF}} \quad (2.1)$$

Výpočet výsledné hodnoty pro diferenční měření ukazuje vztah (2.2). I zde platí, že  $U_{REF}$  je referenční hodnota napětí. Dále zde vystupuje proměnná  $G$ , což znamená zesílení,  $U_{POS}$  je hodnota neinvertujícího vstupu a  $U_{NEG}$  znamená hodnotu na invertujícím vstupu. [3]

$$ADCH:ADCL = G \cdot 512 \cdot \frac{(U_{POS} - U_{NEG})}{U_{REF}} \quad (2.2)$$

Poslední popisovaná funkce mikrokontroléru je jednotka USART. Zjednodušené blokové schéma této jednotky obsahuje tři části a to: generátor hodin, vysílač a přijímač. Generátor hodin může pracovat ve čtyřech režimech. Jsou to: normální asynchronní režim, asynchronní režim s dvojnásobnou rychlostí, synchronní režim master a synchronní režim slave. Pro výběr synchronního, nebo asynchronního režimu slouží bit UMSEL v registru UCSRC. Možnost použití dvojnásobné rychlosti řídí bit U2X v registru UCSRA. Výběr přenosové rychlosti je řízen pomocí zápisu do jednoho 16bitového registru UBRR, který je rozdělen na dva 8bitové registry UBRRH a UBRRL. Výpočet přenosové rychlosti pro normální asynchronní režim ukazuje následující vztah (2.3), kde  $f_0$  je hodinový kmitočet mikrokontroléru.

$$f_{BR} = \frac{f_0}{16 \cdot (UBRR + 1)} \quad (2.3)$$

Ze vztahu (2.3) lze odvodit vztah pro výpočet nastavení registru UBRR (2.4).

$$UBRR = \frac{f_0}{16 \cdot f_{BR}} - 1 \quad (2.4)$$

Pro použití symbolové rychlosti 9600 Bd, musí být nastaven registr UBRR na hodnotu sto tři.

Jednotka USART dovoluje různá nastavení přijatého znaku. Formát datového rámce obsahuje jeden start-bit, pět až devět datových bitů, žádnou paritu, nebo jeden paritní bit a rámec ukončuje jeden, nebo dva stop bity. Pro nastavení délky znaku slouží bity UCSZ0 až UCSZ2 v registrech UCSRB a UCSRC. Paritu lze nastavit bity UPM0 a UPM1 v registru UCSRC. Poslední volba počtu stop bitů je provedena pomocí bitu USBS v registru UCSRC.

Před samotným přenosem hodnot musí být nejprve jednotka USART inicializována. Při této inicializaci musí být zakázané přerušení. Touto inicializací se nastavuje přenosová rychlost, formát datového rámce a povolení použití přijímače a vysílače.

Pro vyslání znaku je nutné povolit vysílač bitem TXEN v registru UCSRB. Přenos znaku je zahájen zápisem do registru UDR. Při vysílání znaků je používáno několik příznakových bitů. Příznakovým bitem UDRE lze zjistit, zda je datový registr vysílače prázdný a příznakem TXC je signalizováno dokončení vysílání rámce.

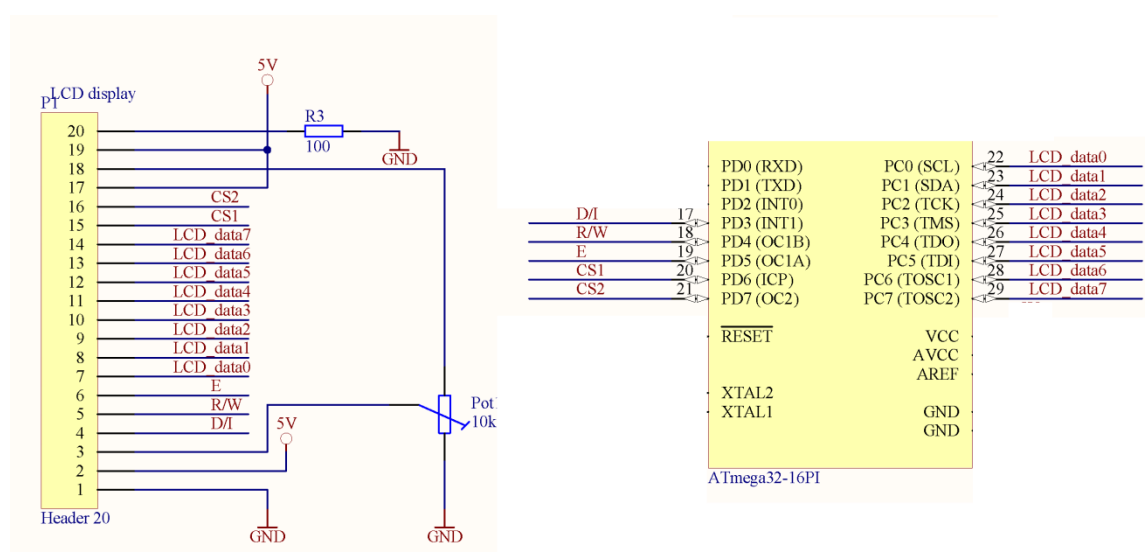
Naopak pro příjem znaku je nutné aktivovat přijímač pomocí bitu RXEN v registru UCSRB. Při příjmu prvního stop-bitu je již celý znak uložen v přijímacím bufferu, který lze číst pomocí registru UDR. Pro příjem znaku lze využít také několik příznakových bitů. Příznakový bit RXC indikuje, že v přijímacím bufferu jsou stále nepřečtené hodnoty. Další příznakové bity signalizují chyby při příjmu. Bit FE značí chybu rámce, bitem DOR se lze dozvědět o ztrátě znaku a bitem PE je signalizována chyba parity. Příznak DOR indikuje ztrátu znaku díky přeplnění vstupního bufferu. Vstupní buffer obsahuje místo pouze pro dva znaky. [3]

## 2.7 LCD displej

Naměřená data lze odečítat především pomocí počítačové aplikace, ovšem může nastat případ, kdy není k dispozici osobní počítač. Z těchto důvodů je zařízení vybaveno

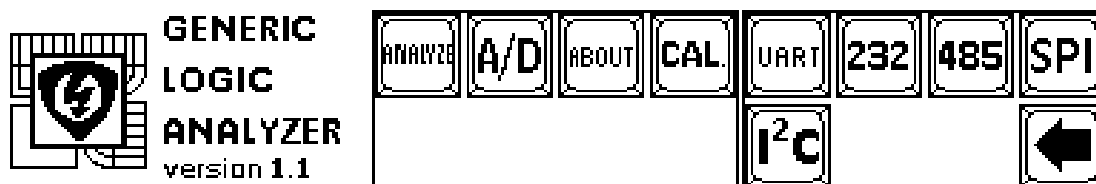
jednobarevným grafickým displejem. Pomocí tohoto displeje lze nejen odečíst naměřená data, ale také ovládat celé zařízení pomocí dotykové fólie.

V zařízení je použitý displej MG12864A – SBC/H, který obsahuje řadič KS0108B. Tento LCD displej disponuje rozlišením 128x64 bodů, což je dostatečné pro pohodlné ovládání zařízení a zobrazení naměřených dat. Ovládání vykreslování dat na displeji řídí mikrokontrolér ATmega32. Nevýhodou tohoto displeje je, že neobsahuje žádnou znakovou sadu. Tu je nutné tedy samostatně vytvořit. Pro připojení displeje je nutné použít osm datových vodičů, což znamená jeden celý port mikrokontroléru a dalších pět řídicích vodičů. Data jsou přenášena paralelní sběrnicí označovanou jako D0 až D7. Řídicí piny jsou D/I, R/W, E, CS1 a CS2. Displej je rozdělen na dva bloky po 64 bodech. Mezi jednotlivými bloky je vybíráno pomocí vodičů CS1 a CS2, všechny ostatní vodiče jsou spojeny paralelně. Pinem R/W je řízeno, zda se data zapisují, nebo jsou čteny. Rozlišení mezi zápisem dat a instrukcí nám zajišťuje pin D/I. Připojení displeje k mikrokontroléru ukazuje obr. 12. [17]



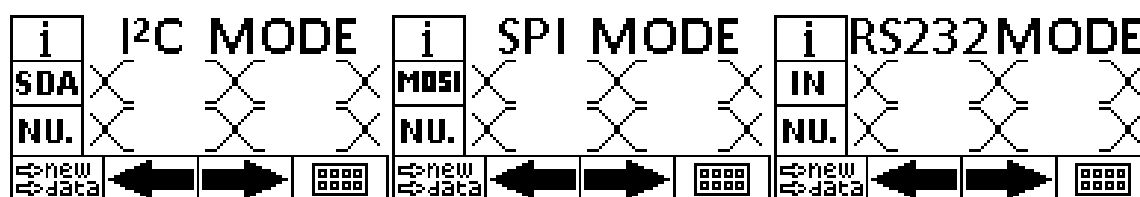
Obr. 12: Připojení grafického displeje k mikrokontroléru ATmega32

Řídicí program pro ovládání displeje je vytvořen v jazyce C. Knihovna, která byla vytvořena přímo pro tento řadič, je stažena od autora Fabian Maximilian Thiele. [18] Tato knihovna obsahuje základní ovládací prvky, jako vykreslení bitmapy, nastavení fontu, vypsání textového řetězce apod. Kvůli speciálním grafickým potřebám logického analyzátoru a také dotykového ovládání bylo nutné tuto knihovnu rozšířit. Rozšíření se týká především možnosti zobrazování ikon, které jsou nutné pro vytvoření ovládacího menu. Vytvořené funkce lze nalézt v souboru TM.c. Bitmapy které jsou zobrazované jako podklady, jsou vytvořené v programu malování přímo na rozlišení displeje a poté převedeny do pole znaků pro jazyk C programem LCDAssistant. Na následujících obrázcích lze vidět úvodní obrazovku, vytvořené hlavní menu a menu pro výběr analyzované sběrnice. Tyto vytvořené menu jsou složené z několika ikon, čili není problém s jejich vzájemným prohozením nebo vypuštěním jen některé z ikon. To přináší komfortnější tvorbu grafického menu.



Obr. 13: Úvodní obrazovka, hlavní menu a menu funkcí analyzátoru

Dále bylo nutné vytvořit uživatelské grafické rozhraní pro interpretaci naměřených dat. Na obr. 14 jsou některé vybrané grafické podklady.



Obr. 14: Grafický design pro I²C, SPI, RS232, RS485, pro funkce A/D a GPIO

Praktickou ukázkou vytvořeného grafického rozhraní přímo na použitém displeji ukazuje obr. 15.

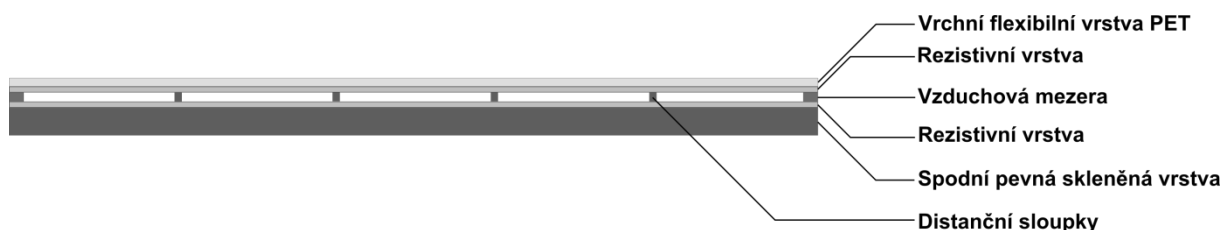


Obr. 15: Praktická aplikace vytvořeného ovládacího menu

## 2.8 Dotyková fólie

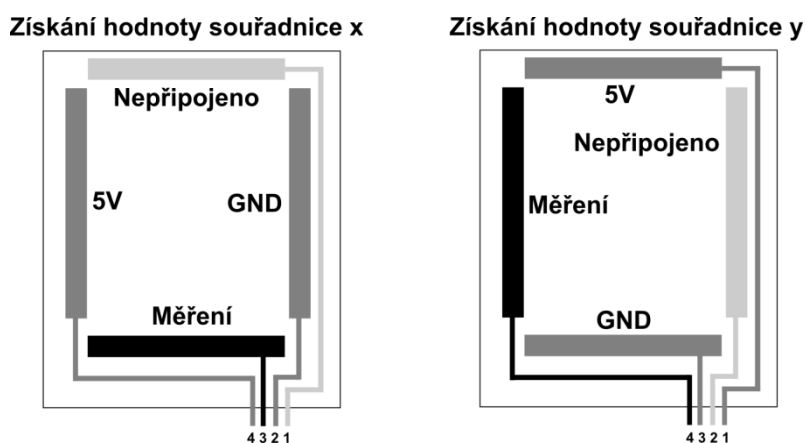
Stejně tak jak je nutné při absenci počítače zobrazit naměřené hodnoty na displeji, tak je nutné ovládat při tomto stavu chod aplikace. To je zajištěno dotykovou fólií, která je umístěna nad displejem. Tímto způsobem ovládání se až na spínač vstupního napětí odstraní mechanické tlačítka a tím i nutnost odstranění zámků.

Dotyková fólie je tvořena ze dvou destiček, které jsou oddělené tenkou vzduchovou mezerou. Jedna destička je pevná a zpravidla je vytvořena ze skla. Druhá ohebná destička je vyrobena z Polyetylenu (PET) a tvoří vrchní část fólie. Obě destičky jsou potaženy tenkou odporovou vrstvou. Často se jedná o sloučeninu nazývanou Indium Tin Oxid (ITO). Ve spodní části tvořené ze skla jsou přidány malé distanční sloupky, které drží ohebnou vrchní fólii, aby nedocházelo k prověšení nebo k detekci falešných dotyků. Pokud je flexibilní horní vrstva stlačena, dojde k dotyku obou odporových vrstev. Místo dotyku určuje velikost odporu, který lze měřit pomocí mikrokontroléru. Struktura dotykové fólie je znázorněna na obr. 16. [4]



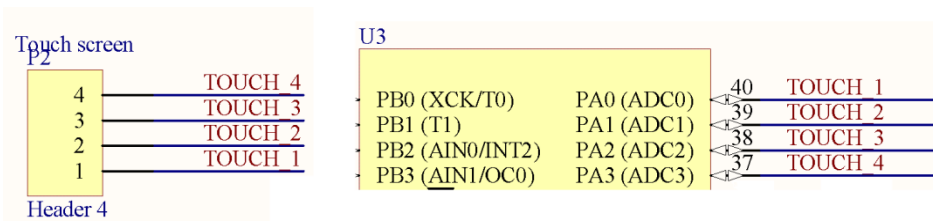
Obr. 16: Konstrukce dotykové fólie, převzato z [4]

Jelikož není tato fólie vybavena žádným čipem pro převod analogových hodnot do digitální reprezentace, je nutné měřit analogovou hodnotu vypovídající o poloze dotyku a tuto hodnotu převést pomocí A/D převodníku. Zapojení a měření analogové hodnoty probíhá dle obr. 17. Jako první se na vývod čtyři přiloží napětí 5 V a na druhý vývod zem. Poté na třetím pinu lze odečítat napětí proti zemi, které je úměrné poloze na ose X. Po odečtení hodnoty přivedeme vstupní napětí na první pin a na druhý přiložíme zem. Čtvrtým pinem odečítáme hodnotu napětí a tím i polohu na ose Y. [4]



Obr. 17: Zapojení vývodů dotykové fólie, převzato z [4]

Připojení dotykové fólie k mikrokontroléru ATmega 32 ukazuje obr. 18.



Obr. 18: Připojení dotykové fólie k mikrokontroléru ATmega32

Při získávání hodnot z dotykové fólie dochází k vzájemnému ovlivňování jednotlivých naměřených hodnot z os X a Y. Tím dochází také i k falešným stisknutím. Tyto stavy jsou ošetřeny pomocí vytvořené funkce `touch_scan()`. Touto funkcí se vždy zjišťuje korektní stisknutí dotykové fólie. Základem funkce je zjišťování hodnoty z A/D převodníku pro osu X. Toto měření proběhne desetkrát. Pokud z deseti vzorků jsou tři po sobě jdoucí stejné velikosti, funkce vrátí jako návratovou hodnotu `log. 1`, v opačném případě `log. 0`. Pokud jsou dva vzorky po sobě jdoucí stejné, ale třetí již ne, vynuluje se počet stejných vzorků a inkrementace začíná opět od nuly.

### 2.8.1 Kalibrace dotykové fólie

Kvůli tomu, že přiložená dotyková fólie nemá stejné rozměry jako použitý grafický displej, je nutné určit okrajové hranice zobrazené oblasti. Pro přesnější odezvu stisku byla vytvořena funkce pro kalibraci dotykové fólie. Při výběru ikony „CAL.“ je možné určit krajní body. V každém rohu se postupně zobrazí měřicí bod, který je potřeba stisknout. Vždy je nutné držet stisk až do doby, než se objeví nový měřicí bod ve vedlejším okraji. Po stisknutí všech čtyř měřicích bodů se uloží nové nastavení do paměti EEPROM mikrokontroléru ATmega32. Toto nastavení je vždy po aktivaci zařízení opět nahráno, tudíž stačí pouze jedna kalibrace. Pro případ chyby kalibrace a uložení špatných hodnot, kde by již hrozilo, že dotyková fólie není schopna správně detekovat stisk a tím by nešla ani spustit nová kalibrace, se nastaví původní předem určené hodnoty.

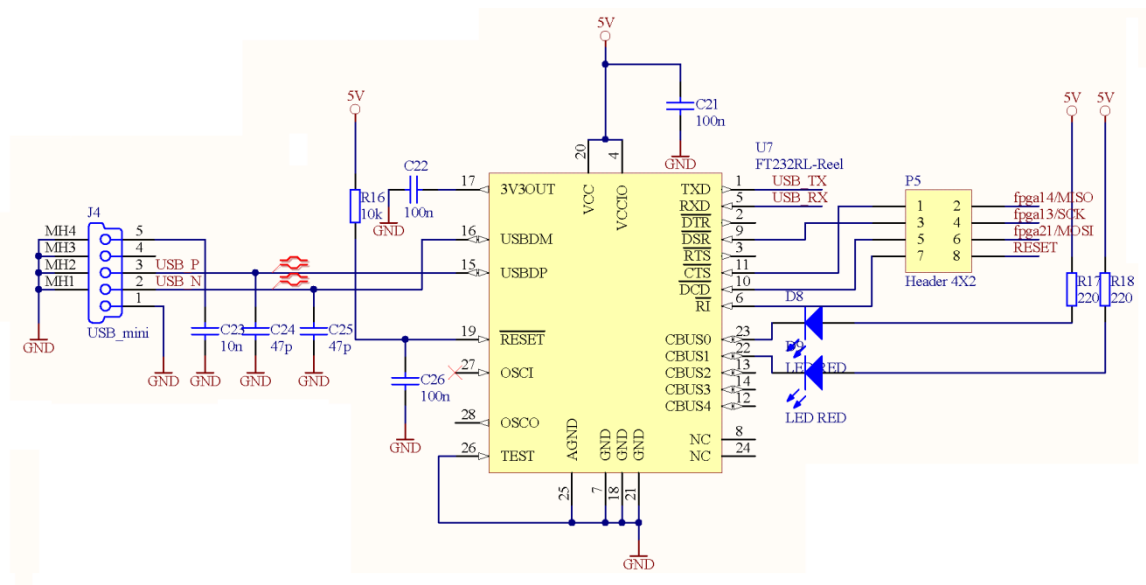
## 2.9 Vstupní bloky A/D a IrDA

Vytvořený analyzátor obsahuje nejen obvody pro logickou analýzu, ale také možnost přímého měření analogových signálů pomocí dvou A/D převodníků. Díky tomu, že jsou tyto dva A/D převodníky multiplexovány a snímání probíhá po 250 ms, trvá změření obou hodnot 500 ms. Časové omezení vzniká především díky funkcím pro zobrazení v reálném čase v osobním počítači. Grafické interpretace pomocí analogových měřičů a následné zobrazení sta předchozích vzorků v grafickém záznamu neumožní rychlejší snímání analogových hodnot. Oba vstupní díly obsahují dělič napětí, díky kterým se maximální vstupní hodnota napětí rovná 50 V. Při použití 8bitového převodníku a při referenčním napětí 5 V se kvantovací krok rovná 19,607 mV.

Další možností měření je použití tohoto analyzátoru pro detekci přenosu pomocí infračerveného záření. Pro měření lze použít např. infračervený senzor TSOP4836.

## 2.10 Komunikace osobního počítače s logickým analyzátozem

Naměřené údaje je nutné přenést do osobního počítače. Tato data jsou přenášena pomocí standardu USB. Komunikaci zajišťuje integrovaný obvod FT232RL, který vytvoří v počítači virtuální sériovou linku, s kterou lze pracovat stejně jako se sériovým portem. Obvodové zapojení, které je nutné pro konverzi z USB na USART, je k vidění na obr. 19.



Obr. 19: Schematické zapojení obvodu FT232RL, převzato z [19]



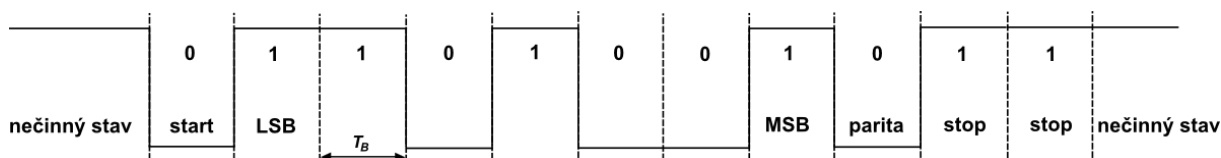
### 3 Popis analyzovaných datových protokolů

Signály, které logický analyzátor identifikuje lze dělit na sériové, paralelní, synchronní a asynchronní. Sériový přenos dat probíhá po jedné nebo pouze po pár datových vodičích, přičemž data jsou posílána v časovém multiplexu. Zato při paralelním přenosu dat probíhá komunikace po více vodičích, aby bylo možné během jednoho časového úseku zaslat celý datový byte. Synchronní signály jsou takové, které svůj stav mění zároveň s hodinovým taktem označovaný jako CLK. Mezi tyto sběrnice patří např. I<sup>2</sup>C a SPI. Zatímco asynchronní obvody hodinový impuls nepoužívají a jejich synchronizace a ošetření chyb probíhá jiným způsobem. Za zmínku stojí např. UART, USB či RS232. Asynchronní komunikace je tvořena pevně danou strukturou a probíhá tedy po datových rámcích. Tyto rámce tvoří z pravidla pět až devět datových bitů, volitelný paritní bit a volba stop bitu, které jsou určeny časem  $T_B$ . Pokud neprobíhá komunikace, jsou datové sběrnice ve stavu vysoké logické úrovně. Komunikace začne probíhat, pokud je nastaven start bit, který je vždy určen stavem log 0. Stop bit, popřípadě stop bity, jsou dány vysokou logickou úrovní. Posledním volitelným bitem v rámci je parita. Tímto bitem je zajištěna ochrana proti chybám. Používá se sudá i lichá parita. Parita určuje počet bitů ve stavu log 1. Výpočet je prováděn logickou funkcí XOR všech bitů dle vztahu (3.1).

$$P_L = d_{MSB} \oplus \dots \oplus d_{LSB} \oplus 1 \quad (3.1)$$

$$P_S = d_{MSB} \oplus \dots \oplus d_{LSB} \oplus 0$$

Zobrazení struktury asynchronního rámce pro režim 7E2, což znamená, že tento rámec obsahuje sedm datových bitů, sudou paritu a dva stop bity, ukazuje obr. 20.



Obr. 20: Struktura asynchronního rámce pro režim 7E2, převzato z [5]

Počet přenesených dat lze určit dvěma způsoby: symbolovou rychlostí a bitovou rychlostí. Symbolová rychlost určuje, jak často může docházet ke změně logické úrovně. Výpočet symbolové rychlosti je určen vztahem (3.2).

$$SR = \frac{1}{\text{čas mezi přechody}} [Bd] \quad (3.2)$$

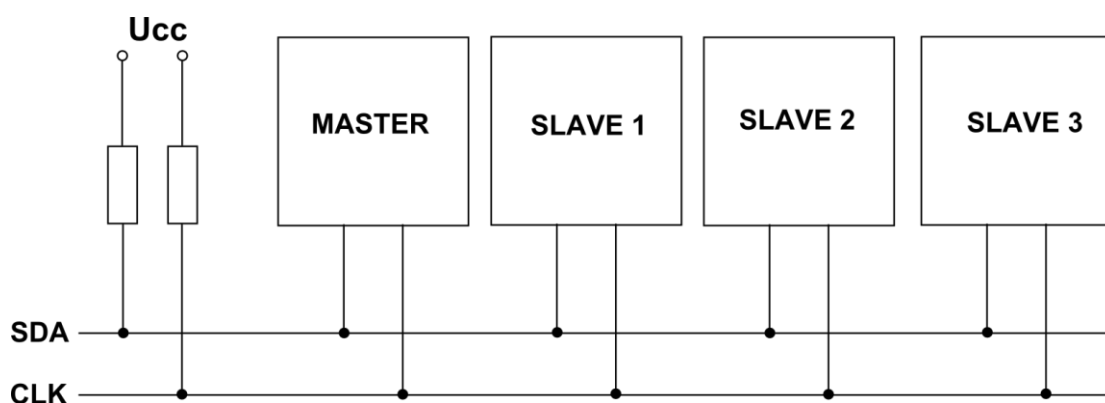
Bitovou rychlostí je určeno, jakou frekvencí jsou přenášeny bity. Výsledné hodnoty bitové rychlosti lze získat dosazením do vztahu (2.3). [5]

$$BR = \frac{1}{T_B} [bit/s] \quad (3.3)$$

V následujícím textu jsou popsány významné datové sběrnice, které je tento univerzální analyzátor schopen zaznamenat.

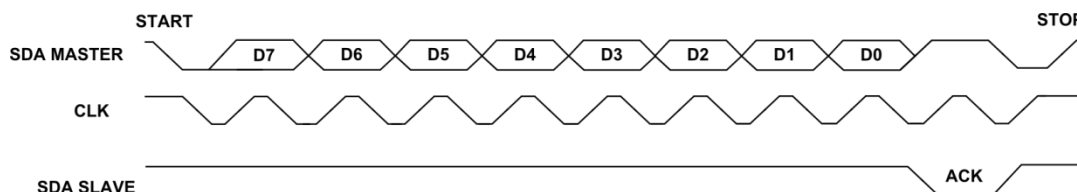
### 3.1 I<sup>2</sup>C

Jedná se o oboustrannou sériovou komunikaci vytvořenou firmou Philips Semiconductors, která bývá někdy také označováno jako TWI. Pro komunikaci mezi zařízeními je použito dvou vodičů. Jeden z nich je použit jako datový vodič a druhý plní funkci hodinového signálu. Datový vodič je označován jako SDA, hodinový signál jako CLK. Na SDA i na CLK musí být připojen zdvihací „pull-up“ rezistor o velikosti několika kilo ohmů. Tímto ochráníme obvody v situaci, kdy by chtěl vysílat zároveň více než jeden obvod. Hodinový signál a obvod, s kterým se bude komunikovat, určuje zařízení označované jako MASTER. Toto zařízení je na sběrnici pouze jedno a ostatní připojené obvody se označují jako SLAVE. Strukturu tří připojených SLAVE obvodů a obvodu MASTER na sběrnici I<sup>2</sup>C lze vidět v obr. 21.



Obr. 21: Propojení jednotlivých zařízení na sběrnici I<sup>2</sup>C

Komunikace mezi jednotlivými zařízeními probíhá v následujících etapách. Jako první MASTER vyšle start-bit označovaný jako S a zahájí komunikaci po sběrnici. Následující osmice bitů obsahuje adresu obvodu a řídicí bit, který je na pozici D0 a určuje směr přenosu. Adresa je pevně zabudovaná v obvodu, ovšem tři bity bývají většinou volitelné, pro případ použití jednoho obvodu vícekrát. Pokud spojení proběhlo úspěšně, potvrdí připojený obvod spojení potvrzovacím bitem, který je značený jako ACK. Po potvrzení spojení obvodu pokračuje komunikace dalšími osmi datovými bity. Komunikace se ukončí stop bitem, označovaným jako P. Start-bit je proveden tehdy, pokud se změní stav na datové lince SDA log 1. do log 0. za předpokladu, že je CLK ve stavu log. 1. Naopak stop-bit určuje přechod linky SDA z log. 0 do stavu log. 1, při vysoké úrovni hodinového signálu. Z těchto podmínek pro řídicí signál plyne, že při komunikaci musí mít datová linka SDA ustálený stav během celého hodinového impulsu, jinak by ho komunikující obvody braly jako signál řídicí start popřípadě stop. Vyslání jednoho bytu a potvrzení přijetí SLAVE obvodem po I<sup>2</sup>C sběrnici znázorňuje obr. 22.



Obr. 22: Zobrazení přenosu jednoho datového bytu pře sběrnici I<sup>2</sup>C, převzato z [6]

Komunikační rámec mezi dvěma zařízeními je znázorněn na obr. 23.



Obr. 23: Zobrazení komunikačního paketu sběrnice I2C, převzato z [6]

Sběrnice I<sup>2</sup>C má definovány dva standardy „fast“ a „standard“. Mezi těmito standardy je rozdíl především v maximální možné rychlosti komunikace. V následující tabulce jsou zobrazeny hlavní parametry obou těchto standardů. [6]

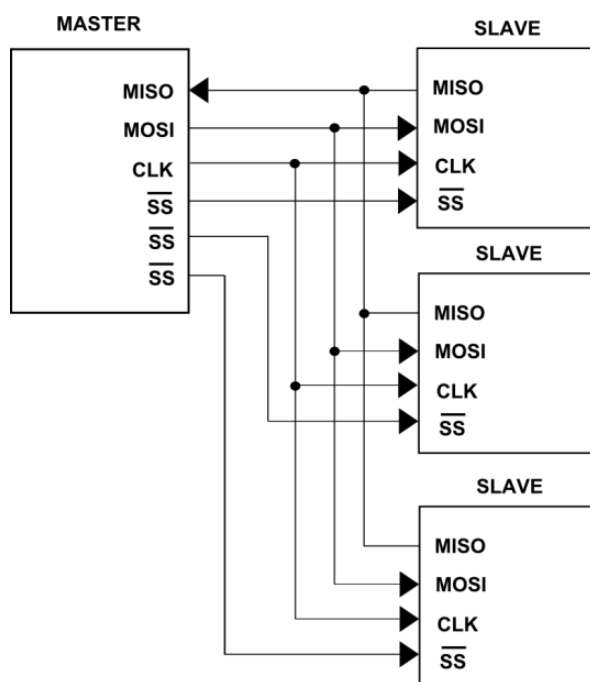
Tab. 5: Parametry standardů I2C, převzato z [6]

Parametr	Standard		Fast		Jednotka
	Min.	Max.	Min.	Max.	
Rozsah vstupních úrovní log. 0	-0,5	1,5	-0,5	1,5	V
Rozsah vstupních úrovní log. 1	3	-	3	-	V
Rozsah výstupních úrovní log. 0	-	0,4	0	0,4	V
Vstupní odběr	-10	10	-10	10	μA
Kmitočet CLK	0	100	0	400	kHz
Doba trvání log. 0 CLK	4,7	-	1,3	-	μs
Doba trvání log. 1 CLK	4,0	-	0,6	-	μs
Doba náběhu	-	1000	-	300	ns
Doba poklesu	-	300	-	300	ns

## 3.2 SPI

SPI – (Serial Peripheral Interface) je sériová synchronní datová sběrnice, která pracuje v plně duplexním režimu. Z toho plyne, že zařízení mohou zároveň vysílat i přijímat data. Tato komunikace probíhá po čtyřech datových linkách a to MOSI – (Master Output, Slave Input), MISO – (Master Input, Slave Output), CLK a SS – (Slave Select). Jak již název napovídá, linka MOSI slouží pro přenos z MASTER obvodu do vybraného SLAVE zařízení. Opačný

směr komunikace zajišťuje signál MISO. Následující signál CLK slouží pro hodinové impulsy a SS slouží k výběru SLAVE zařízení. SLAVE zařízení mají tří-stavové výstupy, které jsou ve stavu vysoké impedance tehdy, pokud není zařízení vybráno ke komunikaci. Komunikace může probíhat mezi dvěma nebo i více zařízeními. Vždy je jedno zařízení MASTER, které ovládá hodinový signál a výběr zařízení, s kterým proběhne komunikace. Pokud je na sběrnici pouze jedno SLAVE zařízení, lze jeho pin SS uzemnit, pokud to použitý obvod dovoluje. Zapojení jednoho MASTER zařízení a tří SLAVE zařízení je vidět na obr. 24.



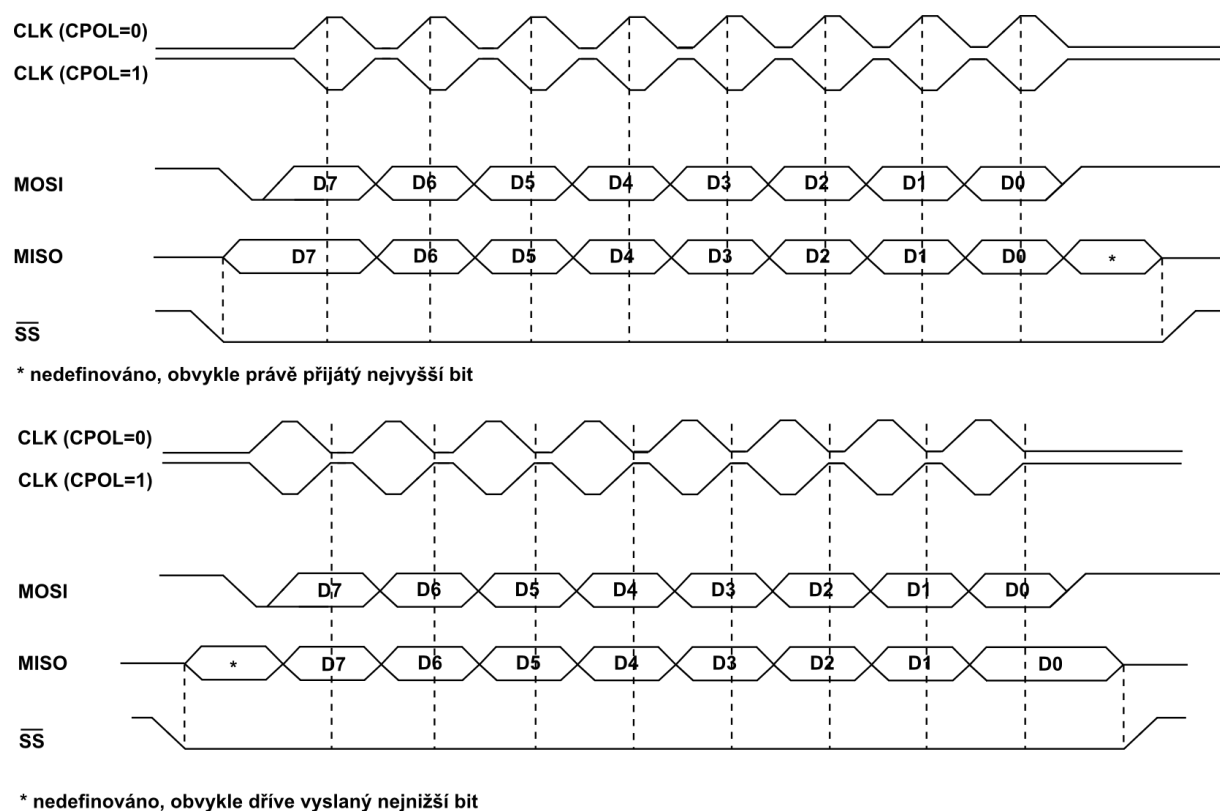
Obr. 24: propojení jednotlivých zařízení na sběrnici SPI, převzato z [7]

Komunikace mezi zařízením MASTER a SLAVE je provedena následujícími kroky. Jako první MASTER nastaví frekvenci hodinových taktů, která se může pohybovat mezi 1–70 MHz. Maximální možnou frekvenci určují možnosti zařízení SLAVE. Dále zařízení MASTER vybere, s kterým zařízením SLAVE bude komunikovat a to tím, že jeho SS nastaví na log. 0. Někdy může být vyžadováno čekání před samotnou komunikací, které MASTER musí dodržovat. Toto je například uplatněno při A/D převodech. Dále jsou s hodinovými impulsy vysílány datové bity ze zařízení MASTER do SLAVE přes MOSI linku a přijímány data v zařízení MASTER ze SLAVE pomocí linky MISO. Vše probíhá v plném duplexním režimu. [7]

Před počátkem komunikace je také nutné nastavit fázi a polaritu přijímání dat v závislosti na hodinovém kmitočtu. Tento popis je znázorněn přímo na použitém mikrokontroléru ATmega32.

Nastavení těchto parametrů lze provést v registru SPCR a to nastavením bitů CPHA a CPOL do stavu log. 1. Pokud je bit CPHA spolu s bitem CPOL roven log. 0 jsou data vzorkována s náběžnou hranou hodinového impulsu. Pokud je ovšem bit CPOL roven log. 1 při log 0 na bitu CPHA, jsou data vzorkována sestupnou hranou hodinového impulsu. Pro stav bitu CPHA, kdy je ve vysoké logické úrovni, platí inverzně předchozí stavy řízení. Bitem

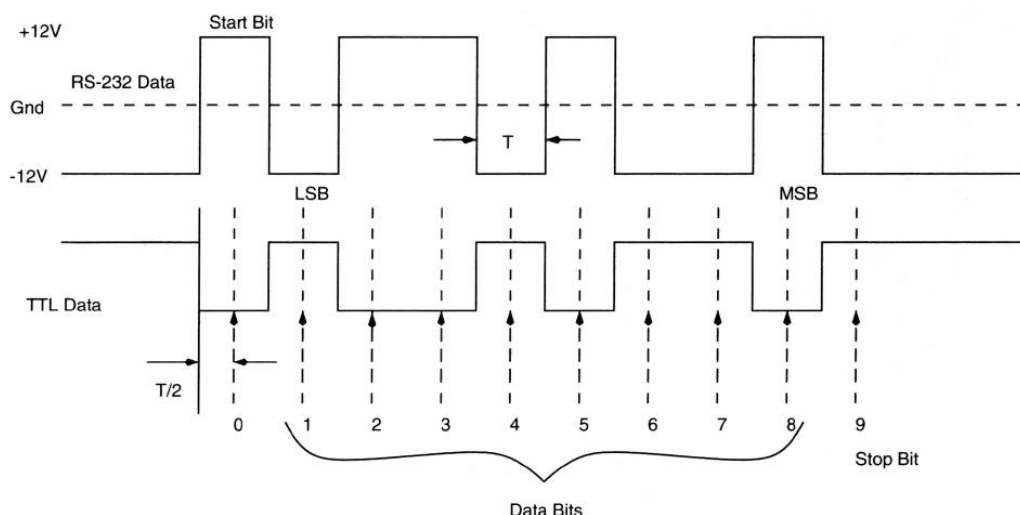
CPOL se také řídí stav CLK při neaktivním přenosu. Při nastavení log. 0 na bitu CPOL je CLK také v nízké úrovni, naopak oproti tomu CPOL ve stavu log. 1 zajistí i vysokou logickou úroveň na pinu CLK. Na obr. 25 jsou jednotlivé situace zakresleny. [3]



Obr. 25: Různé fáze komunikace pře SPI, převzato z [3]

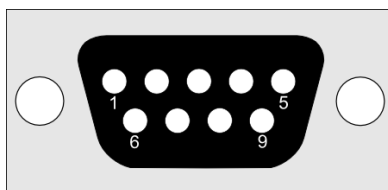
### 3.3 RS232

RS232 je asynchronní standard, který obsahuje bipolární inverzní reprezentaci logických úrovní. Vysoká logická úroveň je reprezentována hodnotou napětí -3 V až -15 V , zatímco nízká logická úroveň je reprezentována hodnotou napětí 3 V až 15 V. V počítačové technice se používají úrovně  $\pm 12$  V, kvůli většímu odolnosti vůči rušení. Na obr. 26. je zobrazena konverze logických úrovní z TTL na RS232. [5]



Obr. 26: Napěťové úrovně standardu RS232 a TTL, převzato z [5]

K připojení zařízení, které spolu komunikují pomocí RS232 jsou používány konektory, Cannon 25, Cannon 9 a RJ45. Nejčastější použití je pomocí konektoru Cannon 9, který je popsán následující tabulkou tab. 6. a obrázkem obr. 27.



Obr. 27: Zapojení konektoru Cannon 9

Tab. 6: Význam jednotlivých pinů konektoru Cannon 9, převzato z [20]

Pin	Název	Směr	Popis
1	CD	←	Carrier Detect
2	RXD	←	Receive Data
3	TXD	→	Transmit Data
4	DTR	→	Data Terminal Ready
5	GND	-	System Ground
6	DSR	←	Data Set Ready
7	RTS	→	Request to Send
8	CTS	←	Clear to Send
9	RI	←	Ring Indicator

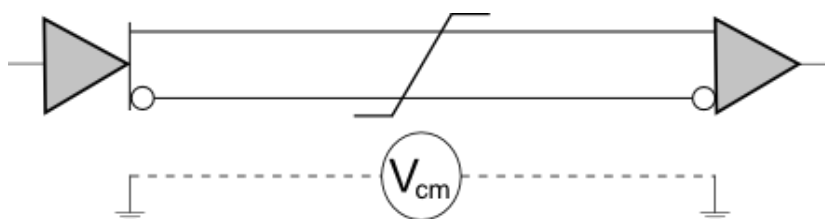
### 3.4 RS485

Jedná se o asynchronní komunikační standard, který se používá především v průmyslu. Na rozdíl od RS232 nejsou jeho logické stavy vyjádřeny hodnotou proti společné zemi, ale jsou vyjádřeny pomocí rozdílového napětí na dvou vodičích označovaných A a B. Díky tomu, že jsou tyto diferenciální vodiče vedeny stejnou cestou a rušení se na nich projeví stejným způsobem, je toto rušení téměř nulové pro rozhodovací úroveň rozdílových signálů. Tím lze dosáhnout delších komunikačních vzdáleností a také větších rychlostí. Běžně se dosahuje rychlostí 2,5 MB/s až 10 MB/s na krátké vzdálenosti, maximální udávaná vzdálenost je kolem 1200 m, pokud je správně provedeno impedanční zakončení a je použita kroucená dvojlinka. Další důležitou vlastností standardu RS485 je, že eliminuje rozdíl zemnicích potenciálů mezi přijímačem a vysílačem. To je v praxi výhodné, pokud jsou systémy napájené jinými napájecími zdroji. [21]

RS485 se používá pro komunikace s více zařízeními, které jsou spojeny jedním signálovým vedením. Většinou je použita architektura MASTER a SLAVE zařízení, přičemž každé komunikující zařízení má vlastní adresu a přijímá pouze pakety, které jsou určeny přímo pro dané zařízení. Pakety generuje MASTER, což může být třeba osobní počítač, který periodicky posílá zprávy všem připojeným SLAVE zařízením. [21]

Komunikace může probíhat po jednom diferenciálním páru, kde je komunikace po tomto páru v obou směrech. Druhá možnost je, že SLAVE zařízení mají vlastní diferenciální pár pro posílání dat do MASTERU a MASTER má opět vlastní diferenciální pár pro zasílání zpráv do SLAVE zařízení. [21]

Pro komunikaci se používá tzv. vyvážený diferenciální pár (Balanced differential pair), kdy vysílač generuje napětí v rozmezí od 2 V až do 7 V mezi výstupy A a B. Zapojení tohoto systému lze nalézt na obr. 28. [21]



Obr. 28: Schematické zapojení Balanced differential páru, převzato z [21]

Napěťová úroveň většiny prodáváných zařízení se pohybuje v rozmezí 0 až 5 V, přičemž v klidovém stavu je na vodiči A 0 V a na vodiči B 5 V. Rozhodovací úroveň rozdílových signálů mezi signály A a B je 200 mV. Přijímač i vysílač jsou spojeny společnou zemí, ovšem ta se nepoužívá pro vyhodnocení stavu na sběrnici. Obvody pro vysílání obsahují povolovací vstupy, které jsou schopny nastavit výstup do stavu vysoké impedance a tím lze na jednu linku připojit několik zařízení. [21]

## 4 Vytvořený subsystém v obvodu FPGA

K vytvoření systému v obvodu FPGA byl zvolen programovací jazyk VHDL a bylo využito programovací prostředí ISE od firmy XILINX. Výsledný zdrojový program je tvořen několika VHDL moduly, které jsou vzájemně spojeny ve schematicém editoru. V následující kapitole je uveden stručný úvod do jazyků HDL, hradlových polí a popis samotného subsystému implementovaného v FPGA.

### 4.1 Programování obvodů FPGA

Programování obvodů FPGA je dnes prováděno nejčastěji dvěma jazyky VHDL a Verilog. Jejich použití lze dělit i geograficky a to tak, že v Evropě je především použit programovací jazyk VHDL, v Asii je více rozšířen Verilog a v USA jsou používány oba programovací jazyky.

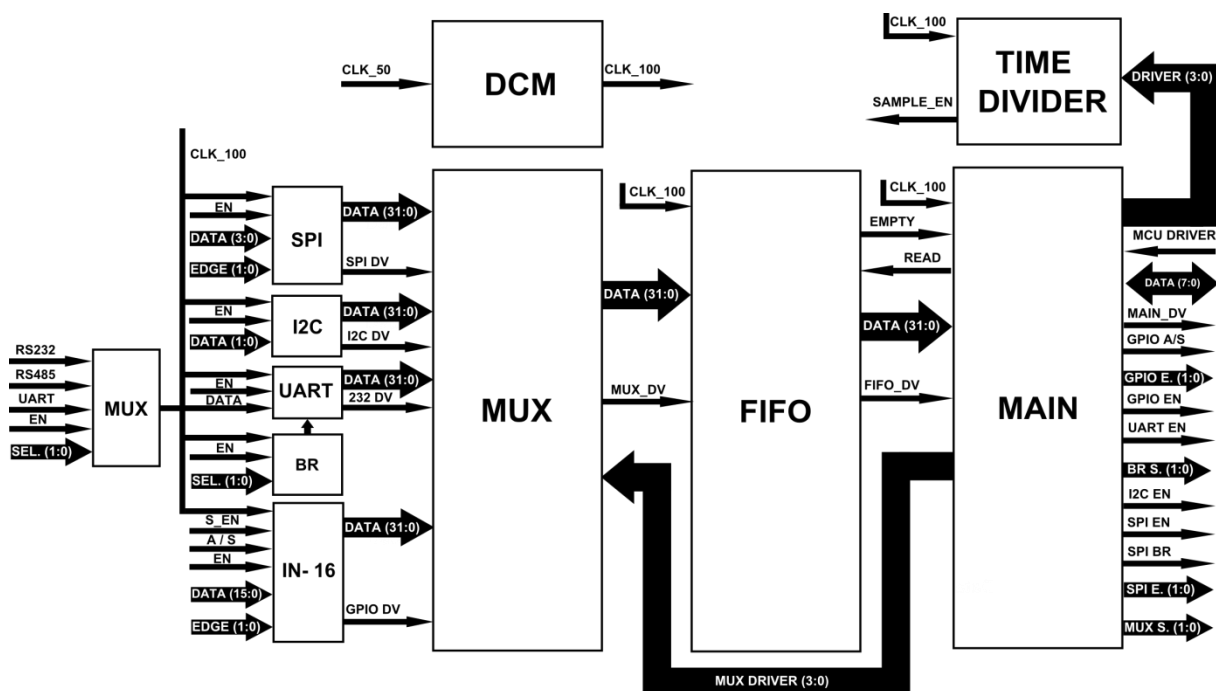
Vývoj jazyka VHDL byl zahájen v roce 1981 ministerstvem obrany Spojených států amerických projektem VHSIC – (Very High Speed Integrated Circuits). Tento projekt měl zajistit popis, návrh a simulaci rozsáhlých číslicových obvodů nezávisle na cílové technologii a návrhu. V roce 1983 byl tento projekt zadán firmám, které se již projektu dříve účastnily. Jednalo se o firmy IBM, Intermetrics a Texas Instruments. Právě tyto firmy v letech 1983 až 1985 daly základ jazyku VHDL, který byl poprvé publikován v roce 1985. V roce 1986 přenechalo ministerstvo obrany standardizaci a další vývoj jazyka na organizaci IEEE – (Institute of Electrical and Electronics Engineers). První publikace jazyku VHDL organizací IEEE byla vydána v roce 1987. Revize jazyku VHDL měla podle zvyklostí organizace IEEE být každých pět let. V dnešní době je všemi systémy podporována verze označovaná jako VHDL-93, i přestože byla vydána v roce 1994. Následné revize v letech 2000 a 2002 se nesetkaly s velikým zájmem firem, proto také nejsou masově rozšířené a nyní se čeká na novou revizi, která by měla přinést velké změny. Pokud se mnohá vylepšení v nové revizi opravdu objeví, lze očekávat větší zájem firem implementovat ji oproti starším revizím.

Programovací jazyk VHDL se používá pro vývoj, popis a simulaci rozsáhlých číslicových systémů. Tímto jazykem přímo popisujeme hardware, proto je nutné myslet na to, že výsledný kód musí být syntetizovatelný přímo pro složení z hradel či klopných obvodů. Výjimku tvoří kódy vytvořené pouze pro simulaci nazvané jako TESTBENCH. [9]

### 4.2 Struktura subsystému FPGA

Výsledný subsystém je složen z následujících modulů: DCM, TIME DIVIDER, SPI, I2C, MUX UART, BR, UART, IN-16, MUX, FIFO a MAIN. Schéma propojení těchto modulů lze vidět na obr. 29.





Obr. 29: Vytvořený subsystém v obvodu FPGA

Prvním blokem DCM (Digital Clock Manager) je zvýšen taktovací kmitočet z původních 50 MHz na pracovní kmitočet 100 MHz. Tímto kmitočtem jsou dále taktovány všechny vytvořené moduly.

Následujícím popisovaným modulem je TIME DIVIDER, který je použit pro možnost zmenšení vzorkovací frekvence pro modul IN-16. Tímto modulem lze získat následující vzorkovací frekvence: 100 MHz, 50 MHz, 25 MHz a 1 MHz. Získání těchto frekvencí je dosaženo pomocí metody „clock enabling“. Výběr vzorkovací frekvence v modulu TIME DIVIDER řídí modul MAIN na základě přijatého požadavku z vytvořené počítačové aplikace.

Blok MAIN zajišťuje obousměrnou komunikaci s mikrokontrolérem ATmega32 a řídí také chod celého subsystému v obvodu FPGA. Ke komunikaci s mikrokontrolérem je využita obousměrná 8bitová sběrnice a řídicí signál MCU DRIVER. Pokud je řídicí signál ve stavu nízké logické úrovně, přijímá modul MAIN řídicí data z mikrokontroléru. V opačném případě jsou posílány do mikrokontroléru nasnímané hodnoty. Tento blok také řídí povolení pro jednotlivé snímací režimy. V modulu IN-16 dále vybírá mezi synchronním a asynchronním režimem a v případě výběru synchronního režimu také přepíná mezi citlivostí na náběžnou hranu, nebo na sestupnou hranu. Pro snímání standardů RS232 a RS485 navíc blok MAIN provádí nastavení symbolové rychlosti a pro modul SPI vybírá, zda bude snímání citlivé na náběžnou, nebo sestupnou hranu. Dále lze tyto bloky nastavit do režimu spuštění podmíněného příjmem specifického rámce, nebo do režimu, ve kterém započne snímání bez ohledu na obsahu zachyceného rámce. Další funkcí modulu MAIN je výběr datové cesty pomocí vytvořeného multiplexoru v modulu MUX v závislosti na povolené vstupní periférii. Aby bylo možné poslat platné nasnímané hodnoty mikrokontroléru ATmega32 je nutné o tyto hodnoty požádat vytvořený blok FIFO, který je řízen na základě uložených dat v této paměti.

Pro ukládání nasnímaných hodnot vysokou frekvencí a docílení následného zpracování a předání hodnot mikrokontroléru podstatně nižší frekvencí je nutné použít vyrovnávací paměť typu FIFO. Základem tohoto modulu je přímo vytvořené IP jádro od firmy XILINX. Tato

paměť obsahuje 2048 pozic pro 32b datová slova. Zápis a vyčtení hodnoty z paměti FIFO probíhá následujícím způsobem. Po zjištění vysoké logické úrovně na signálu MUX\_DV, která musí být pouze jeden hodinový takt, je uložena hodnota ze sběrnice DATA. Po zpracování této hodnoty v paměti, což může trvat další tři hodinové takty, je změněn logický stav na signálu EMPTY z nízké logické úrovně na vysokou. Tento signál je zpracován v bloku MAIN, který zažádá o přečtení uložené hodnoty a to zápisem log. 1 na signál READ. Tato změna vyvolá v modulu FIFO přesun načtené hodnoty na výstupní sběrnici FIFO paměti a zvýšení ukazatele na uložené hodnoty.

Dalším vytvořeným blokem MUX je řízena datová cesta od jednotlivých periférií k paměti FIFO. Aktuální výběr je řízen pomocí bloku MAIN.

Prvním popisovaným modulem, který slouží jako vstupní snímací část, je modul IN-16. Tento modul zpracovává 16bitovou sběrnici v synchronním, nebo asynchronním režimu. Dále lze u synchronního režimu vybrat, zda se mají data ukládat při vzrůstající nebo klesající úrovni hodinového signálu.

Z důvodů použití sériového rozhraní pro komunikaci mezi mikrokontrolérem ATmega32 a osobním počítačem muselo být zvoleno řešení pro snímání asynchronní sběrnice, které nebude vyžadovat ke komunikaci minimálně stejnou, nebo vyšší přenosovou rychlost jako je vzorkovací frekvence analyzátoru. Proto je v analyzátoru vytvořena hardwarová komprese hodnot při výběru asynchronního režimu. To znamená, že není posílán aktuální stav sběrnice s každou vzrůstající hranou vybrané vzorkovací frekvence, ale pouze pokud dojde ke změně hodnoty na jakémkoliv pinu sběrnice oproti předchozímu stavu. Při této změně je přenášena hodnota 16bitové sběrnice a také je k této hodnotě navíc přidán aktuální stav vnitřního čítače, který informuje o době trvání předchozího stavu. Ovšem pokud je následující stav sběrnice stejný, dojde pouze k inkrementaci vnitřního 16bitového čítače. Pokud se tento vnitřní čítač dostane na svoji maximální hodnotu, přenesou aktuální stav sběrnice i pokud nenastala změna v celém časovém intervalu. V následující tabulce tab. 7 lze odečíst minimální a maximální možnou délku snímání pro všechny hodnoty vzorkovací frekvence a nastavení požadovaného počtu 2048 platných dat. Skutečná hodnota snímaného času závisí na rychlosti změn stavů na sběrnici.

Tab. 7: Minimální a maximální možný dosažený nasnímaný čas pro různé vzorkovací frekvence

$f_{vz}$ [MHz]	T [ns]	Minimální snímaný čas [ms]	Maximální snímaný čas [s]
<b>100</b>	10	0,04	1,34
<b>50</b>	20	0,08	2,68
<b>25</b>	40	0,16	5,36
<b>1</b>	1000	4,10	134,2

Pokud je vybraný synchronní režim snímání, tak jsou data ukládána s hodinovým signálem, který musí být přiveden na pin IN-1. Vnitřní čítač je nastaven na konstantní hodnotu a z toho plyne, že při tomto režimu nelze odečíst časovou závislost signálů.

Dalším vstupním blokem je modul, který zpracovává asynchronní sběrnice protokolů UART, RS232 a RS485. Tento modul je složen se tří částí a to: modul pro vytvoření

symbolové rychlosti, modul pro zachycení vyslaného znaku a multiplexoru, který vybírá mezi vstupními piny pro připojení jednotlivých standardů. Modul pro výběr symbolové rychlosti je řízen blokem MAIN, který určuje zadanou symbolovou rychlost. Tato symbolová rychlost může nabývat hodnot 4800, 9600, 19200 a 57600 Bd. V tomto režimu není do aplikace přenášén údaj o časové značce, protože zobrazení probíhá pomocí rámců.

Následující popisovaný modul zpracovává hodnoty podle standardu I<sup>2</sup>C. Po detekci start bitu jsou posléze zachytávána data a po příjmu stop bitu je celý znak vyslán do paměti FIFO opět bez časové značky. Kvůli tomuto omezení nejsme schopni rozlišit, jestli se jedná o standard I<sup>2</sup>C Fast, nebo Standard. Funkci detekce by bylo možné doplnit přidáním pomocného čítače, který by měřil periodu hodinového signálu na základě známé frekvence hlavního hodinového singálu FPGA.

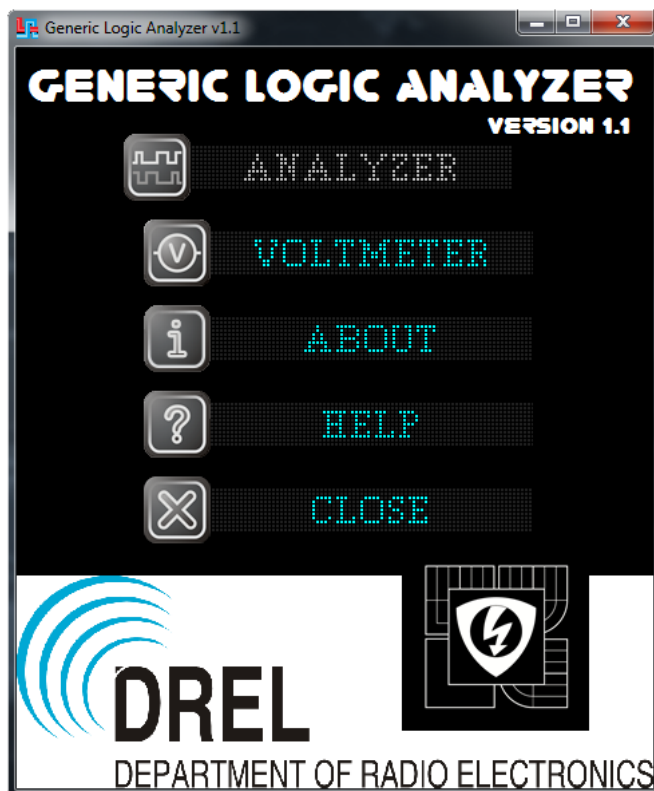
Posledním vstupním blokem je blok SPI. Tímto blokem lze snímat oba datové vodiče MOSI i MISO. V tomto modulu nejsou přijaté hodnoty rámcovány do přijatého znaku. Díky tomu, že jsou posílány stavy dvou vodičů, je posílána vždy změna signálu synchronně s CLK\_SPI. Následné rámcování provádí až mikrokontrolér, nebo aplikace v osobním počítači.

## 5 Vytvořená počítačová aplikace

Hlavním zobrazovacím a ovládacím zařízením je osobní počítač, proto je nutné buďto stáhnout a implementovat již hotový program vhodný pro tyto účely, nebo tuto aplikaci vytvořit. V této práci je zvoleno druhé řešení, kdy je vytvořena aplikace přímo pro toto zařízení.

### 5.1 Hlavní ovládací menu

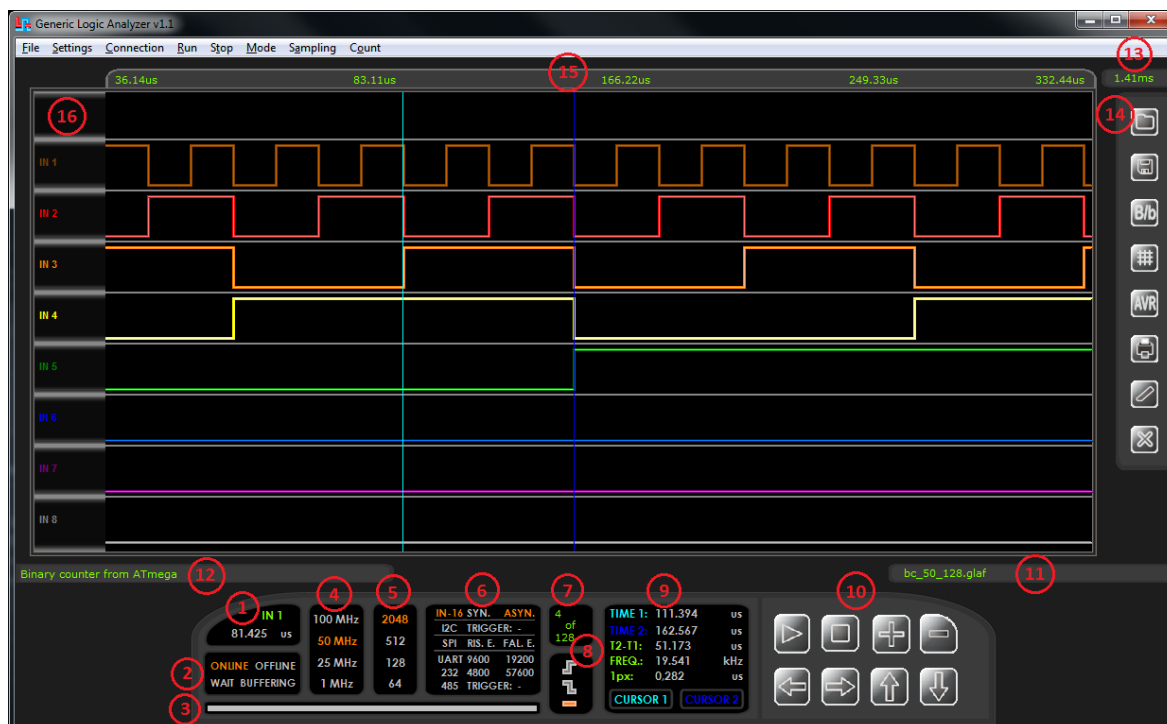
Aplikace je vytvořena pomocí programu C++ BUILDER . Základní struktura programu je následující. Jako první se zobrazí hlavní menu, které obsahuje funkce: logický analyzátor, voltmetr s grafickým záznamem dat, zobrazení informací o aplikaci, nápověda a ukončení aplikace. Grafické ztvárnění hlavního menu aplikace ukazuje obr. 30.



Obr. 30: Hlavní menu vytvořené aplikace, aktuální výběr funkce analyzátoru

### 5.2 Grafický návrh logického analyzátoru

Po výběru funkce analyzátoru, se zobrazí vytvořená aplikace pro interpretaci nasnímaných hodnot. Hlavním prvkem vytvořené aplikace je plocha pro vykreslení nasnímaných hodnot a dva ovládací bloky. Na obr. 31 lze vidět vytvořenou aplikaci.



Obr. 31: Grafické rozhraní vytvořeného analyzátoru

Vytvořený spodní ovládací panel obsahuje deset částí, které jsou rozdělené podle funkcí. Panel číslo 1 zobrazuje aktuální pozici kurzoru v čase a název vybraného signálu. Pomocí panelu číslo 2 se lze připojit nebo naopak odpojit k analyzátoru pomocí sériového rozhraní. Také se lze z tohoto panelu dozvědět, jestli aplikace teprve čeká na platné hodnoty, nebo naopak už započalo ukládání nasnímaných dat. Průběžný poměr obdržených dat oproti požadovanému počtu nasnímaných hodnot ukazuje posuvný ukazatel v panelu číslo 3. Následujícími panely číslo 4 lze vybrat vzorkovací frekvenci a číslem 5 lze nastavit počet požadovaných hodnot. Panelem číslo 6 je možné vybrat režim snímání, pokud jsou dostupné různé volby pro daný režim tak i jeho modifikaci. Panel číslo 7 slouží k informaci o posunutí prvního zobrazeného bitu a celkovém počtu přijatých hodnot. Dalším panelem číslo 8 je vybírána volba spouštění pro režim IN-16. Lze volit mezi možnostmi zachycení na sestupnou hranu, vzestupnou hranu, nebo vypnutí zachycení, které je aktuální pro asynchronní režim snímání. Poslední informační panel číslo 9 je používán pro zobrazení kurzorů, k odečtení časového úseku mezi nimi a následnému přepočtu na frekvenci. Další informaci, kterou lze zde získat je maximální rozlišení kurzoru na jeden pixel. Panelem číslo 10 je ovládáno zobrazení získaných hodnot a také spouštění, nebo naopak přerušení přenosu nasnímaných nových hodnot.

Následují čtyři samostatné části číslo 11, 12, 13 a 15. V části číslo 11 se zobrazuje název nahraného souboru. V případě, že se nejedná o nahraný soubor, zobrazuje se nápis „New file“. Do panelu číslo 12 lze napsat vlastní poznámku, která je uložena spolu se souborem. Poslední informační panely zobrazují časové vlastnosti naměřeného signálu. Panel číslo 13 zobrazuje celkový nasnímaný čas analyzovaného průběhu a časová osa číslo 15 aktuální zobrazenou časovou oblast.

Druhý ovládací panel číslo 14 slouží k otevření uloženého souboru, uložení snímaného signálu, přepnutí funkce zobrazení, zobrazení mřížky, vygenerování souboru pro

mikrokontroléru AVR, tisku zobrazeného signálu, vymazání aktuálních hodnot a k zavření aplikace. Tyto funkce lze ovládat také pomocí vytvořeného menu v horní části aplikace.

Poslední označená část programu číslo 16 je zobrazení nasnímaných hodnot a k umístění kurzorů.

## **5.3 Vytvořené funkce logického analyzátoru**

Vytvořený analyzátor disponuje základními funkcemi k zobrazení naměřených hodnot, ke správě naměřených hodnot, k volbě snímacího režimu a několika doplňkovými funkcemi.

### **5.3.1 Volba snímacího režimu**

Aplikace podporuje změnu nastavení vzorkovací frekvence, změnu počtu požadovaných hodnot, změnu snímacího režimu a jeho rozšiřující nastavení pokud ho daný režim obsahuje.

Vzorkovací frekvence může nabývat hodnot: 100 MHz, 50 MHz, 25 MHz a 1 MHz. Změnu vzorkovací frekvence lze provést kliknutím na požadovanou hodnotu ve vytvořeném panelu číslo 4, nebo pomocí vytvořeného menu pomocí volby „Sampling“ a poté výběrem vzorkovací frekvence. Aktuální výběr se v panelu označí oranžovou barvou a v menu se také označí zaškrtnutím.

Počet požadovaných hodnot lze měnit v rozmezí 64 až 2048. Změnu lze provést pomocí volby v panelu číslo 5, nebo pomocí volby „Count“. Aktuální výběr se v panelu označí oranžovou barvou a v menu se také označí zaškrtnutím.

Poslední možností lze měnit mezi režimy: IN-16, I2C, SPI, UART, RS232 a RS485. Dále pro režim IN-16 lze volit mezi synchronním, nebo asynchronním režimem. Při synchronním režimu lze pomocí panelu číslo 8 vybrat, jestli bude prováděno snímání se vzestupnou nebo sestupnou hranou hodinového signálu. Pro režim SPI lze zvolit synchronizaci na vzestupnou nebo sestupnou hranu a pro režimy UART, RS232 a RS485 je k dispozici na výběr ze čtyř možností symbolové rychlosti. Tyto volby lze provést pomocí panelu číslo 6, nebo pomocí volby v menu „Mode“. Dále lze v panelu číslo 6 vybrat pro režimy UART, RS232, RS485 a I2C spouštění na zadané číslo. Po stisknutí nápisu TRIGGER se objeví dialogové okno, kde se zadává požadovaná hodnota, která ovšem musí být v rozmezí 0 – 255 dekadicky. Po opětovném stisku se deaktivuje volba spouštění na číslo. Pokud je spouštění aktivováno a uživatel chce změnit číslo, které je zadané, stačí kliknout na předchozí číslo a opět se objeví dialogové okno se zadáním nového spouštěcího znaku.

### **5.3.2 Použití kurzorů**

Vytvořená aplikace disponuje základní možností měření časových údajů pomocí kurzorů. Aktivaci a posléze i deaktivaci jednotlivých kurzorů lze provést pomocí volby „CURSOR 1“ a „CURSOR 2“ v panelu číslo 9. Po zobrazení kurzorů je lze ovládat levým a pravým tlačítkem myši. Kurzor 1 je ovládán levým tlačítkem myši a kurzor 2 pravým. Po výběru požadovaného časového úseku lze odečíst časy jednotlivých kurzorů, rozdíl mezi těmito časy a také přepočítanou odpovídající frekvenci. Poslední informací je přesnost měření na jeden zobrazený pixel. Tento údaj se mění na základě změny časového měřítka.

### 5.3.3 Ovládání snímání a zobrazení oblasti

Pro možnosti ovládání snímání a zobrazení bylo vytvořeno osm tlačítek v panelu číslo 10. Tlačítko „play“ slouží ke spuštění snímání a k zaslání nastavení veškerých voleb v obvodu FPGA. Tlačítkem „stop“ lze přerušit načítání hodnot a zobrazit pouze doposud přijaté data. Tlačítka „plus“ a „mínus“ je zvětšována nebo naopak zmenšována zobrazená oblast. Šípkami vpravo a vlevo lze posouvat první zobrazený bit v zobrazené oblasti. Šípkami nahoru a dolů je přepínáno při výběru režimu IN-16 mezi zobrazením hodnot prvních osmi pinů sběrnice, nebo druhé osmice pinů. Pro pohodlnější práci se zobrazenými údaji lze místo tlačítek vlevo, vpravo, plus a mínus využít i kolečko myši, kdy při současném držení klávesy Ctrl lze zvětšovat nebo zmenšovat zobrazenou část a při použití bez klávesy Ctrl lze měnit první zobrazenou hodnotu.

### 5.3.4 Ukládání a nahrávání nasnímaných hodnot

Pro možnost nahrávání a ukládání nasnímaných hodnot byly vytvořené funkce, které lze ovládat pomocí panelu číslo 14 (ikona složky a diskety), nebo pomocí volby v menu „File“ a poté „Open“, nebo „Save as“. Vytvořený soubor obsahuje příponu „.glaf“ (Generic logic analyzer file). Při otevření souboru pomocí textového editoru, lze vidět uložení v podobě sloupce textových údajů. První uložený údaj obsahuje počet uložených hodnot. Tento údaj je čtyřnásobkem zadaných požadovaných hodnot, protože toto číslo obsahuje 16bitové data a také 16bitový časový údaj, to znamená čtyři 8bitové hodnoty. Dále jsou uloženy platné data v počtu, který závisí na první hodnotě. Následující hodnoty určují snímaný režim, vzorkovací frekvenci, nastavení pro daný režim, informaci o vybrané hraně synchronizace, celkový čas a vytvořenou poznámku. Vzorový uložený projekt lze vidět v následující tabulce tab. 8.

Tab. 8: Ukázka uloženého souboru

Číslo řádku	Hodnota	Popis
1	512	Počet uložených dat
2	0	Hodnota sběrnice 9 - 16
3	2	Hodnota sběrnice 1 - 8
4	2	Časovač 15b – 8b
5	14	Časovač 7b – 0b
513	125	Časovač 7b – 0b
514	1	Režim – IN-16
515	2	Vzorkovací frekvence – 50MHz
516	2	Možnost – Asynchronní režim
517	3	Spouštění - žádné
518	1.41ms	Celkový čas
519	Binary counter	Poznámka

### 5.3.5 Vytvoření zdrojového kódu pro mikrokontrolér řady AVR

Při použití režimu IN-16 a výběru asynchronního snímání lze využít funkci, která vygeneruje zdrojový kód pro mikrokontrolér řady AVR, který se pak může chovat jako zdroj repliky zachyceného signálu. Vytvořený generátor obsahuje prvních osm nasnímaných pinů, které jsou v aplikaci označené jako „IN 1“ až „IN 8“. Tyto snímané signály převede jako výstupní data pro PORTB v mikrokontroléru. Vytvořený vzorový zdrojový kód pro binární čítač lze vidět níže.

```
// Generator for PORTB PB0 = Binary 1
// Created by: Generic Logic Analyzer
// VUT FEEC DREL - Diploma thesis 2013
// Note: binary counter
// Date: 11.4.2013
// Time: 23:23:18

#include <avr/io.h>
#define F_CPU 16000000UL // This must be for delay
#include <util/delay.h>

int main(void)
{
    DDRB = 0b11111111; // Port B is Out

    while(1)
    {
        PORTB = 0b00000110;
        _delay_us(12);
        PORTB = 0b00000111;
        _delay_us(12);
        PORTB = 0b00001000;
        _delay_us(12);
        PORTB = 0b00001001;
        _delay_us(12);
        PORTB = 0b00001010;
        _delay_us(12);
        PORTB = 0b00001011;
        _delay_us(12);
        .
        .
        .
        PORTB = 0b01101111;
        _delay_us(12);
        PORTB = 0b01110000;
        _delay_us(12);
    }
}
```

K vytvoření tohoto zdrojového kódu do textového souboru lze využít ikonu „AVR“ v panelu číslo 14, nebo také pomocí menu přes „File“ a poté „Generate ATmega file“.

Z důvodů použití vygenerovaného kódu na mikrokontroléru ATmega32 je nutné brát v potaz některé limitující vlastnosti. Díky nejmenšímu možnému zpoždění funkce `_delay_us`, které je 1  $\mu$ s, lze dosáhnout pouze frekvencí s maximální periodou 2  $\mu$ s, to znamená, že maximální vytvořená frekvence pomocí tohoto generátoru může být 500 kHz. Dále také kvůli rozlišovací schopnosti 1  $\mu$ s nelze vygenerovat přesnou kopii nasnímaného signálu, která může dosahovat přesnosti 10 ns při použití vzorkovací frekvence 100 MHz. Při generování signálu dochází k odstranění desetinné části, proto může nastat časový rozdíl až 0,999  $\mu$ s. U signálů,



kteře mají menší časovou značku než 1  $\mu$ s, nedojde k vytvoření příslušné změny. Při rozpoznání takových signálů během vytváření zdrojového kódu je posléze uživatel o této skutečnosti varován pomocí dialogového okna, které lze vidět na následujícím obrázku obr. 32.



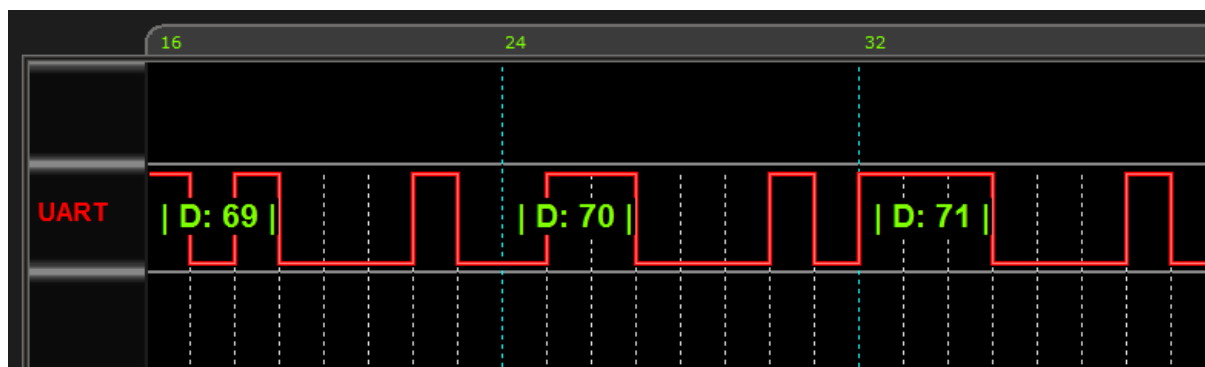
Obr. 32: Informace o nepřesném vytvoření nasnímaného signálu

### 5.3.6 Volby zobrazení

Pro zobrazení nasnímaných hodnot aplikace nabízí možnosti: zobrazení po bitech, kde je délka bitu závislá na čase, zobrazení po bitech, kde je zobrazení všech bitů stejné velikosti a zobrazení po bytech. Pro asynchronní snímání IN-16 je nastaveno automaticky zobrazení po bitech, se zobrazením v závislosti na čase. Aktuální délka zobrazeného bitu je vypočítávána na základě přijaté časové značky. Výpočet zahrnuje přijatou časovou značku ( $t_c$ ), vzorkovací frekvenci ( $f_{vz}$ ), aktuální zobrazovanou délku průběhu ( $t_r$ ) a také aktuální velikost zobrazované plochy v pixelech ( $l_p$ ). Výpočet, podle kterého je zobrazeno kolik pixelů bude vykresleno, ukazuje následující vztah.

$$l = \frac{(t_c + 1) \cdot \left(\frac{1}{f_{vz}}\right)}{\frac{t_r}{l_p}} [px] \quad (5.1)$$

Pro ostatní režimy je přednastavena volba zobrazení bitů s jednotnou délkou a do toho je navíc vložena hodnota celého přijatého slova. Důvodem je to, že po sériovém rozhraní není přenášena každá hrana snímaného signálu, ale změny jsou už zpracovány v obvodu FPGA do podoby přijatého bytu a až poté přenášeny do mikrokontroléru a následně do osobního počítače. Tím ztratíme možnost sledovat časové změny. Pro sériové rozhraní musí být rychlost předem známá a uživatel ji musí vybrat, proto není informace o trvání signálu nutná, ale pro rozhraní I<sup>2</sup>C nejsme schopni rozlišit, jestli se jedná o standard Standard, nebo Fast. Názornou ukázkou zobrazení pro režim UART se zapnutou mřížkou lze vidět na následujícím obrázku obr. 33.



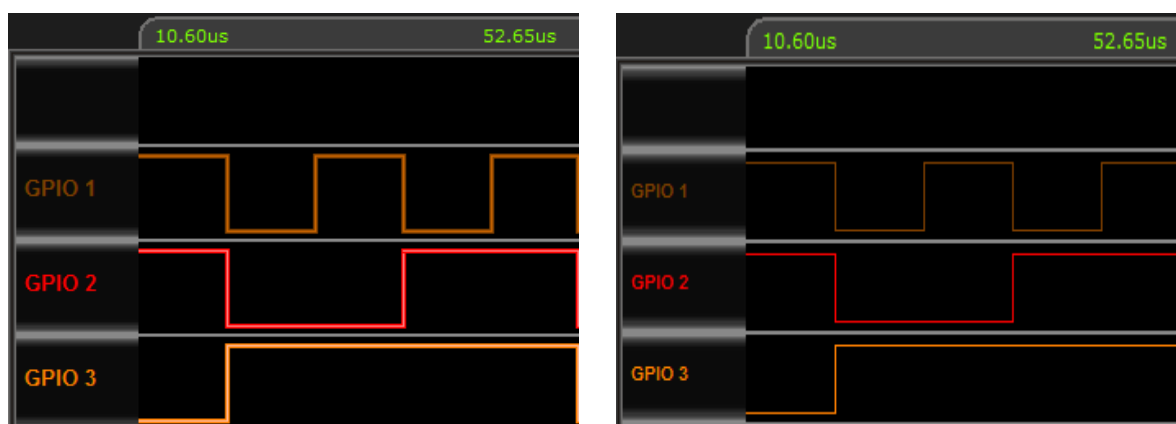
Obr. 33: Zobrazení nasnímaných hodnot pro režimy UART, RS232, RS485. I²C a SPI

Dále lze vybrat zobrazení po bytech, kde se nezobrazí rozklad na jednotlivé bity ale pouze hodnota v dekadickém zobrazení.

Tyto volby lze provést pomocí volby v menu „Settings“ a poté „Shows“, nebo lze přepínat mezi režimy bitového a bytového zobrazení pomocí panelu číslo 14 ikonou „B/b“.

Další volitelnou možností je aktivace, nebo deaktivace mřížky, která je určena pouze pro bitové zobrazení stejné velikosti. Rozděluje názorně jednotlivé bity a po osmi bitech je barevně rozlišen konec přijatého znaku. Tuto volbu lze provést pomocí ikony mřížky v panelu číslo 14 anebo pomocí „Settings“ a poté „Grid“.

Poslední grafická funkce je automatická změna velikosti zobrazení průběhu signálu. Pro menší velikost spuštěné aplikace, nebo větší počet zobrazených hodnot je automaticky provedeno zmenšení zobrazení pouze na jeden pixel ze stávajících tří. Ukázka obou zobrazení je na obr. 34.



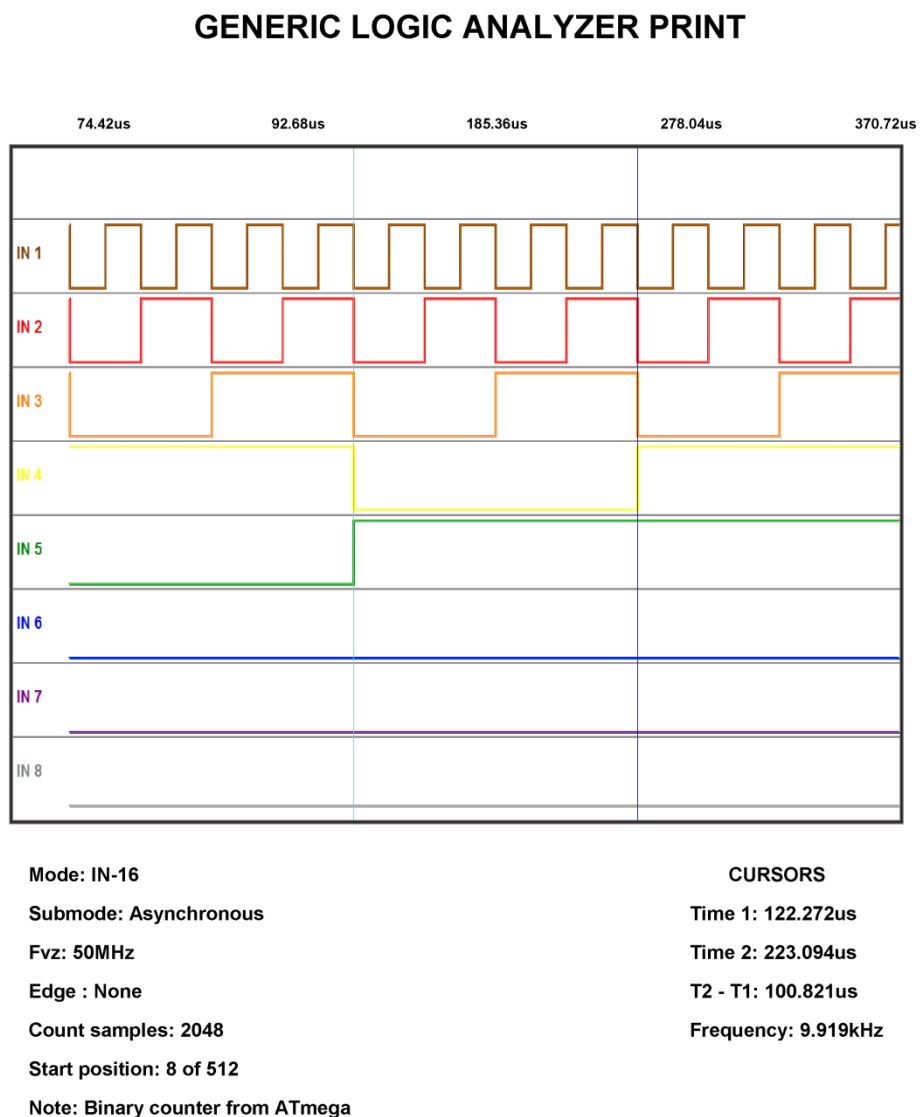
Obr. 34: Ukázka automatické úpravy zobrazení nasnímaného průběhu

### 5.3.7 Tisk

Aplikace disponuje možností vytisknout aktuální zobrazenou oblast, vlastnosti snímání, měření pomocí kurzorů a také základní informace o čase, datu apod. Zobrazená oblast

nasnímaného signálu, je vždy přepočítána na velikost stránky, která je orientována na výšku. Proto je vždy lepší provádět tisk, pokud je aplikace roztažena na větší plochu. Tisk lze provést pomocí ikony tiskárny v panelu číslo 15, nebo pomocí „File“ a poté „Print“.

Výslednou vytištěnou stránku ukazuje následující obrázek obr. 35.



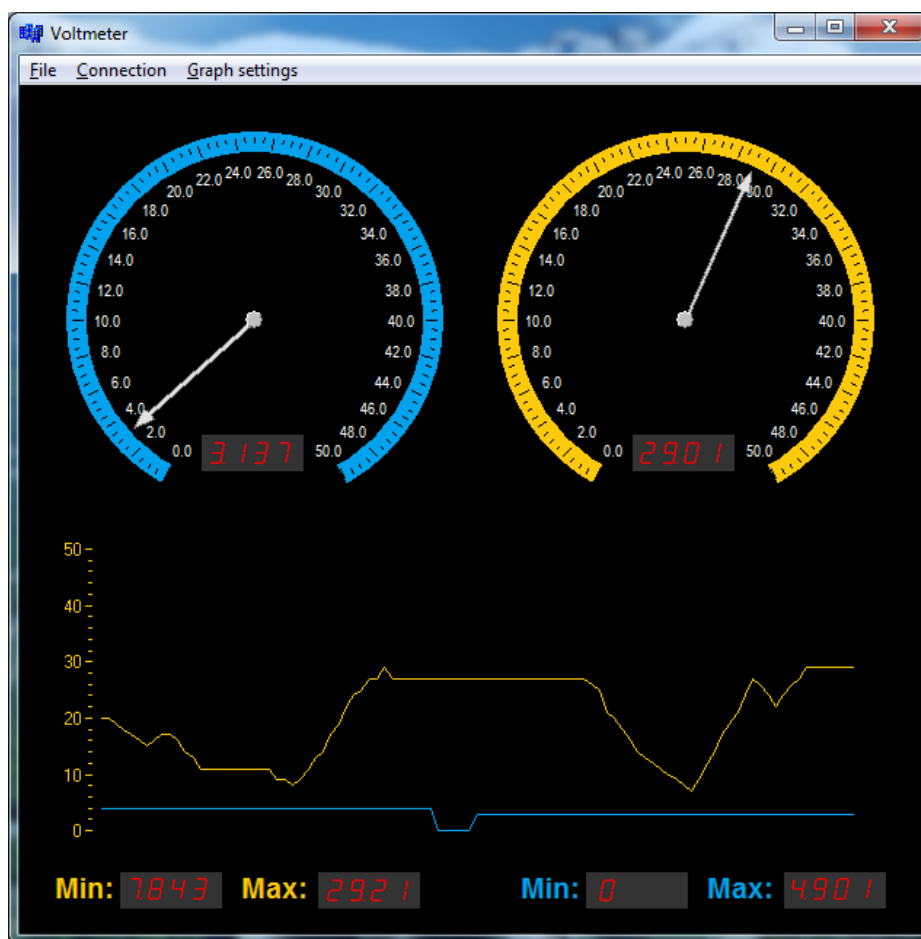
Obr. 35: Zobrazení vytisknutého formuláře

## 5.4 Vytvoření vlastní sběrnice

Vytvořená aplikace umožňuje seskládat vlastní až 8bitovou sběrnici ze vstupních pinů IN1 až IN8. Vytvořená sběrnice může obsahovat třeba pouze dva piny a jednotlivé váhy pinů si lze libovolně nastavit. Vytvořenou sběrnici lze zobrazit pomocí vytvořeného menu přes „Settings“ a poté pomocí „Bus“ a „Enabled“. Změnu pořadí pinů vytvořené sběrnice lze provést přiřazením jiného vstupního pinu k danému pinu sběrnice.

## 5.5 Voltmetr

Další možností, jak využít vytvořenou aplikaci, je použití analyzátoru jako dvojitého voltmetru v rozmezí napětí 0 – 50 V. Získané hodnoty napětí jsou zobrazeny pomocí kruhových zobrazovačů a také ve spodní části zobrazovače pomocí sedmisegmentových displejů. Vytvořený modul také disponuje možností zobrazení posledních sta záznamů pomocí grafu pro oba voltmetry. Vytvořený modul voltmetru lze vidět na obr. 36.

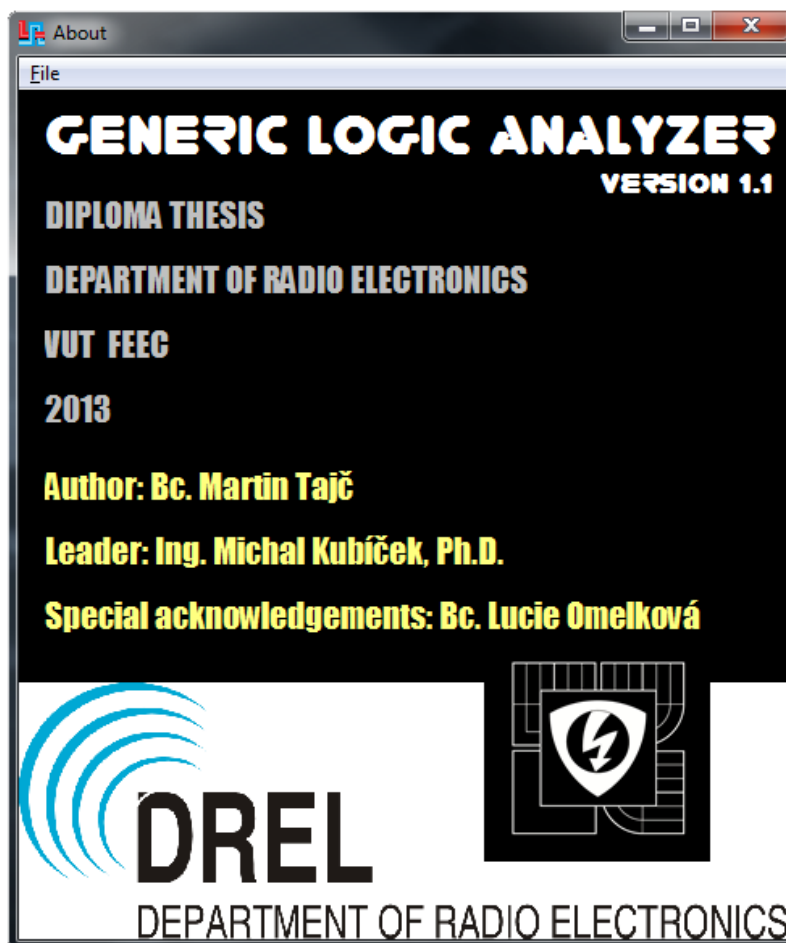


Obr. 36: Zobrazení aplikace pro měření dvou hodnot napětí a jejich grafický záznam

## 5.6 O aplikaci

Jako poslední byl vytvořen modul, který informuje o základních údajích vytvořeného programu. Obsahuje informace, o jaký projekt se jedná, o tom kdy a kde tento projekt vznikl a

také osoby, které se na tomto projektu podílely. Názornou ukázkou informací o aplikaci ukazuje obr. 37.



Obr. 37: Vytvořený modul pro zobrazení informací o aplikaci

## 5.7 Ovládání pomocí klávesových zkratek

Některé volby vytvořeného programu lze také ovládat pomocí klávesových zkratek. Seznam klávesových zkratek, které lze použít, ukazuje následující tabulka.

## Aplikace logického analyzátoru:

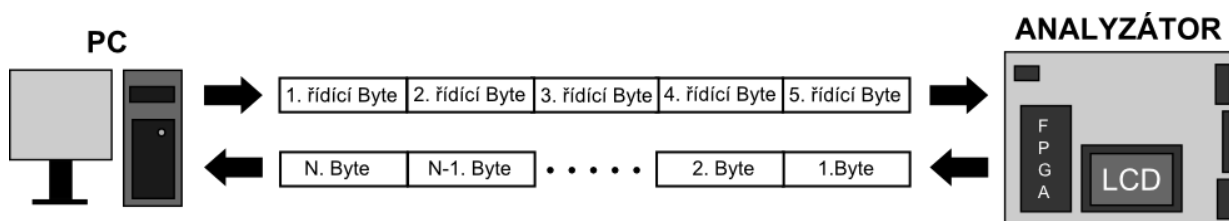
Tab. 9: Seznam použitelných klávesových zkratk v aplikaci analyzátoru

<b>Funkce</b>	<b>Klávesová zkratka</b>
<b>Uložení souboru</b>	Ctrl + S
<b>Otevření souboru</b>	Ctrl + O
<b>Vytvoření zdrojového kódu pro AVR</b>	Ctrl + A
<b>Tisk</b>	Ctrl + P
<b>Konec</b>	Ctrl + X
<b>Zobrazení bytů</b>	Ctrl + B
<b>Zobrazení všech bitů stejnou velikostí</b>	Ctrl + E
<b>Zobrazení všech bitů v závislosti na čase</b>	Ctrl + T
<b>Připojení analyzátoru k zařízení</b>	Ctrl + C
<b>Odpojení analyzátoru od zařízení</b>	Ctrl + D
<b>Režim IN-16</b>	F1
<b>Režim I<sup>2</sup>C</b>	F2
<b>Režim SPI</b>	F3
<b>Režim UART</b>	F4
<b>Režim RS232</b>	F5
<b>Režim RS485</b>	F6
<b>Vzorkovací frekvence 100 MHz</b>	Ctrl + F1
<b>Vzorkovací frekvence 50 MHz</b>	Ctrl + F2
<b>Vzorkovací frekvence 25 MHz</b>	Ctrl + F3
<b>Vzorkovací frekvence 1 MHz</b>	Ctrl + F4
<b>Počet požadovaných hodnot 2048</b>	Shift + F1
<b>Počet požadovaných hodnot 512</b>	Shift + F2
<b>Počet požadovaných hodnot 128</b>	Shift + F3
<b>Počet požadovaných hodnot 64</b>	Shift + F4

## 5.8 Komunikace osobního počítače s vytvořeným hardwarem logického analyzátoru

Osobní počítač komunikuje s vytvořeným hardwarem pomocí rozhraní RS232. K vytvoření spojení mezi aplikací a analyzátozem je nutné nejprve vyslat znak, který má hodnotu jedna. Tento znak je vyslán pouze jednou pro první připojení. Dále je také nutné při vypnutí aplikace

odpojit analyzátor od osobního počítače pro možnost ovládání pomocí grafického displeje. K odpojení je vyslán aplikací znak o hodnotě dva. Jelikož je nutné nastavit spouštění, vzorkovací frekvenci, počet přijatých dat, režim a jeho vlastnosti, byl vytvořen komunikační protokol, který obsahuje pět znaků. Aplikace vysílá jednotlivé znaky s rychlostí 9600 Bd jedním stop bitem a bez parity. Význam jednotlivých řídicích znaků lze nalézt v následujících tabulkách. Na obr. 38 je zobrazena komunikace mezi osobním počítačem a analyzátozem.



Obr. 38: Způsob komunikace osobního počítače s analyzátozem

1 řídicí znak:

Tab. 10: Význam prvního řídicího bytu

Dekadická hodnota	Význam
50	Spouštění na vzestupnou hranu pro synchronní režim IN-16
51	Spouštění na sestupnou hranu pro synchronní režim IN-16
52	Vypnuta spouštění na hranu
53	Spouštění na zadanou hodnotu v rozmezí 0 - 255

2 řídicí znak:

Tab. 11: Význam druhého řídicího bytu

Dekadická hodnota	Význam
20	Vzorkovací frekvence 100 MHz
21	Vzorkovací frekvence 50 MHz
22	Vzorkovací frekvence 25 MHz
23	Vzorkovací frekvence 1 MHz
Vlastní znak	Spouštění na zadanou hodnotu, pouze při režimu spouštění na vlastní hodnotu

3 řídicí znak:

Tab. 12: Význam třetího řídicího bytu

Dekadická hodnota	Význam
<b>30</b>	2048 požadovaných hodnot
<b>31</b>	512 požadovaných hodnot
<b>32</b>	128 požadovaných hodnot
<b>33</b>	64 požadovaných hodnot

4 řídicí znak:

Tab. 13: Význam čtvrtého řídicího bytu

Dekadická hodnota	Význam
<b>40</b>	Synchronní spouštění pro režim IN-16
<b>41</b>	Asynchronní spouštění pro režim IN-16
<b>42</b>	Symbolová rychlost 4800 Bd
<b>43</b>	Symbolová rychlost 9600 Bd
<b>44</b>	Symbolová rychlost 19200 Bd
<b>45</b>	Symbolová rychlost 56000 Bd

5 řídicí znak:

Tab. 14: Význam pátého řídicího bytu

Dekadická hodnota	Význam
<b>10</b>	Režim 16bitové sběrnice
<b>11</b>	Režim I2C
<b>12</b>	Režim SPI
<b>13</b>	Režim RS232
<b>14</b>	Režim RS485
<b>15</b>	Režim UART



Po příjmu těchto pěti znaků se stejné hodnoty předají dál do obvodu FPGA. Po odeslání řídicích znaků aplikace čeká na příjem hodnot. Příjem se ukončí až při příjmu čtyřnásobku požadovaných znaků, protože jsou z obvodu FPGA přeposlány vždy ještě doplňující informace např. o časové značce. Mikrokontrolér tyto znaky opět vysílá symbolovou rychlostí 9600 Bd jedním stop bitem a bez parity. Při dodržení tohoto komunikačního protokolu lze využít vytvořenou aplikaci i k vlastním zařízením.

## 6 Vytvořený program pro mikrokontrolér ATmega32

K vytvoření zdrojového kódu pro mikrokontrolér ATmega32 bylo nutné vybrat způsob programování a také vývojové prostředí, které bude použito. Nakonec bylo zvoleno programování v jazyce C a použití vývojového prostředí AVR Studia.

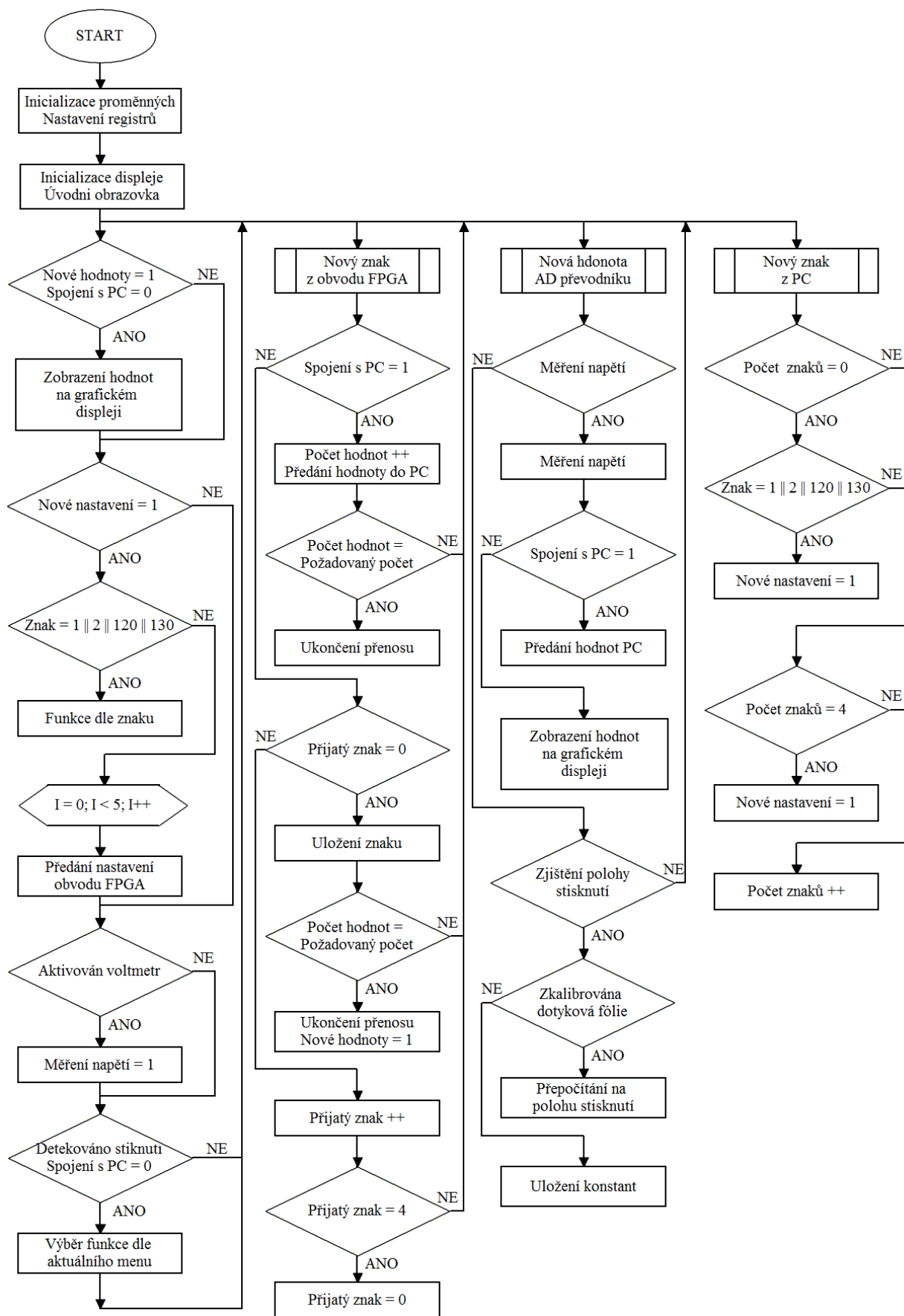
### 6.1 Programování mikrokontroléru

Programovat mikrokontroléry lze dvěma způsoby a to: v jazyku symbolických adres – JSA a v programovacím jazyku C. V jazyku symbolických adres probíhá programování přímo danými instrukcemi konkrétního mikrokontroléru, oproti vyššímu programovacímu jazyku C, kdy je program tvořen univerzálním kódem. Tento kód je přenositelný i mezi dalšími mikrokontroléry. Program vytvořený v programovacím jazyku C je zpravidla přehlednější, ale zato výsledný kód zabírá více místa v paměti. To je způsobeno překladem z vyššího programovacího jazyka do jazyka symbolických adres, který není vždy optimální a vkládá mnohdy zbytečné instrukce. Pokud je potřeba znát přesnou souslednost daných instrukcí, lze do programu vytvořeného v jazyce C vložit také část programu v jazyku symbolických adres. Výhody vyšších programovacích jazyků převyšují jejich nevýhody, proto se dnes používá především toto řešení.

První standardizace jazyka C byla provedena americkou společností ANSI – (American National Standards Institute) v 80. letech 20. století. Následně byl tento jazyk převzat organizací ISO – (International Organization for Standardization), ale i tak zůstal zachován název ANSI C. [8]

### 6.2 Základní vývojový diagram

Vytvořený zdrojový kód lze zobrazit pomocí následujícího základního vývojového diagramu na obr. 39.



Obr. 39: Vývojový diagram vytvořeného programu pro mikrokontrolér ATmega32

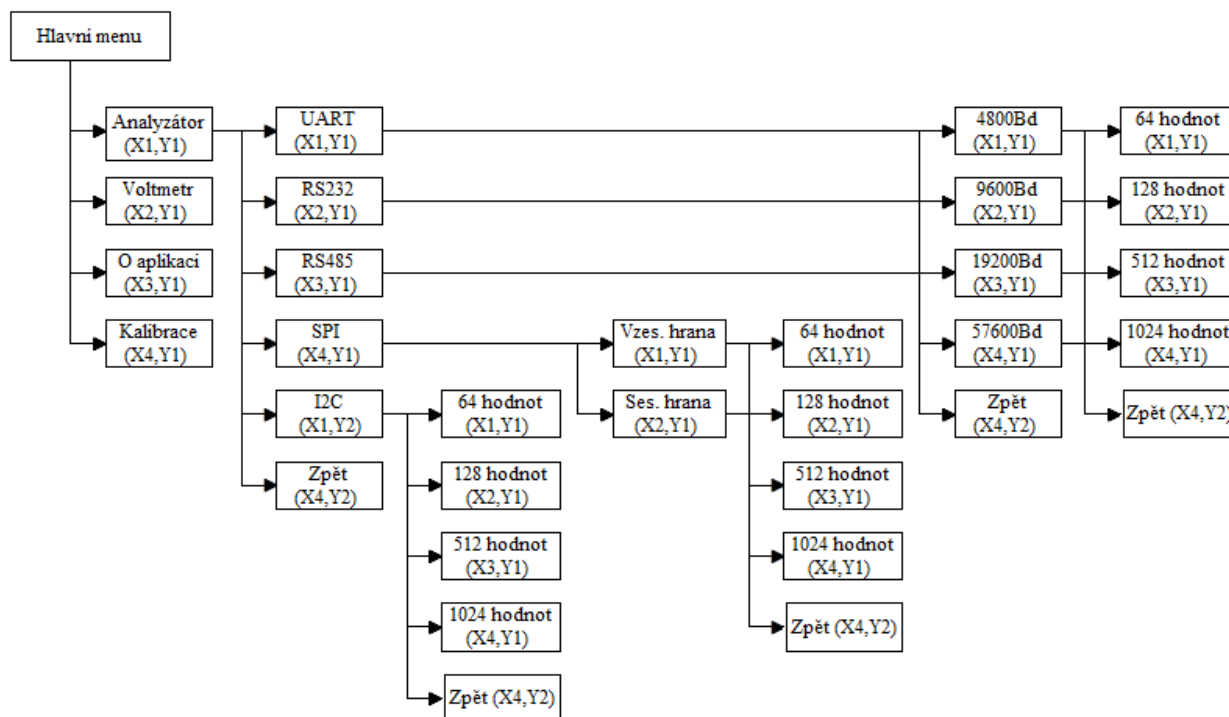
Z vytvořeného vývojové diagramu je patrné, že program obsahuje v nekonečné smyčce několik funkcí jako, zpracování nového nastavení analyzátoru, měření napětí a zjištění stisknutí dotykové fólie. Dále vývojový diagram obsahuje tři přerušení. První přerušení obsluhuje příjem znaků pomocí sériového rozhraní, dále je zde externí přerušení, které informuje o nové hodnotě na sběrnici s obvodem FPGA a poslední přerušení je vyvoláno po dokončení A/D převodu.

Dále byl vytvořen diagram pro větvení vytvořeného menu v závislosti na poloze stisknutí. Jednotlivé umístění polohy stisknutí dotykové fólie lze vidět na obr. 40. Ve vytvořeném diagramu chybí funkce zobrazení pro analyzování 16bitové sběrnice. To je způsobeno obtížností zobrazení časových průběhů pomocí rozlišení grafického displeje.

X1,Y1	X2,Y1	X3,Y1	X4,Y1
X1,Y2	X2,Y2	X3,Y2	X4,Y2

Obr. 40: Rozložení matice pro výběr stisknuté oblasti

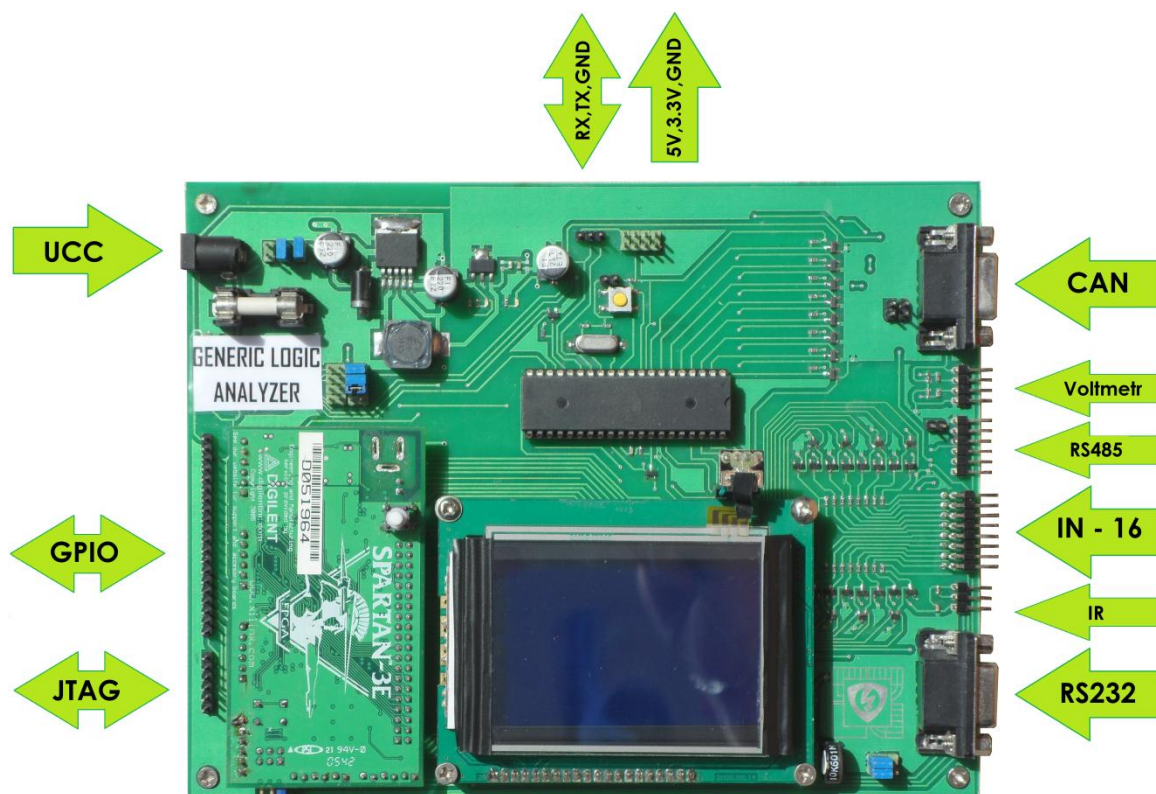
Vytvořený diagram větvení ukazuje obrázek obr. 41.



Obr. 41: Diagram větvení vytvořeného menu pro grafický displej

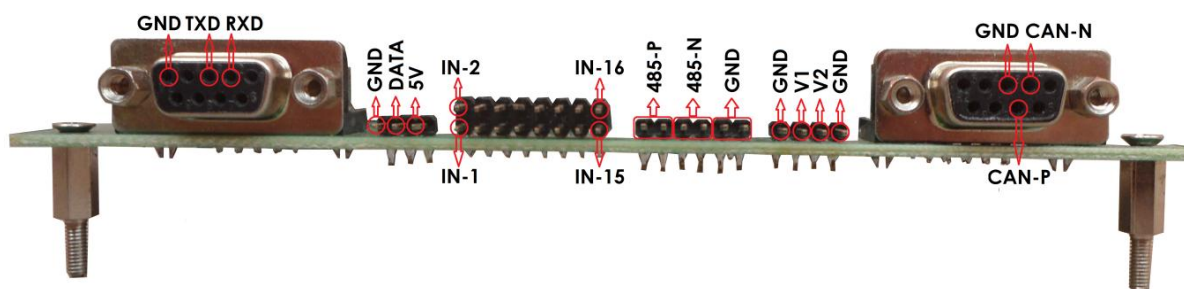
## 7 Popis připojení analyzátoru

Vytvořený analyzátor disponuje několika konektory pro připojení externích periférií, napájecím konektorem, konektorem pro připojení komunikace s osobním počítačem, rozšiřujícími piny k obvodu FPGA, konektorem pro připojení rozhraní JTAG a konektorem s dvěma napětími 5 V a 3,3 V. Na obr. 42 lze vidět rozmístění konektorů na desce plošných spojů.



Obr. 42: Rozmístění konektorů na desce plošných spojů

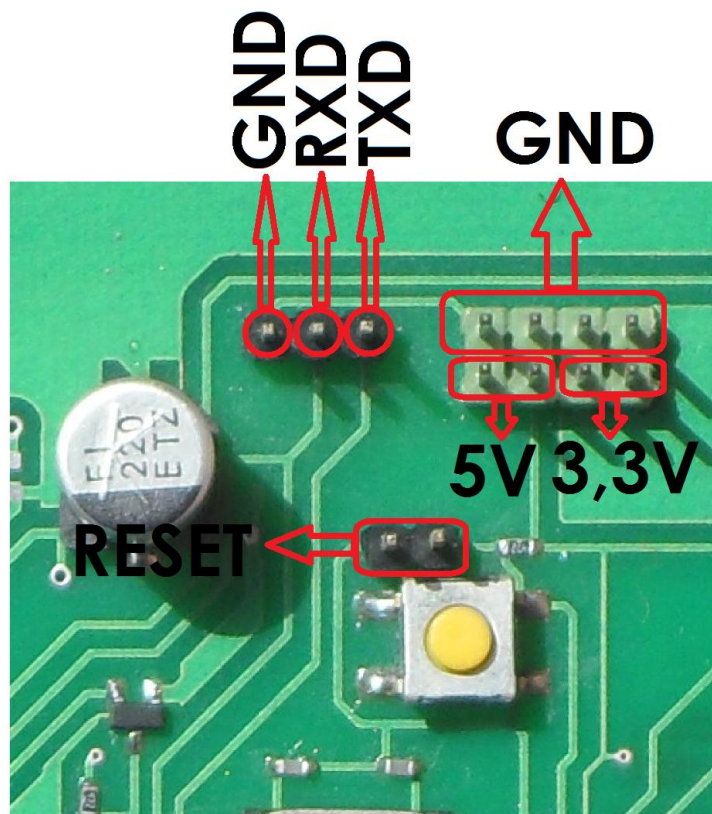
## 7.1 Popis konektorů pro připojení analyzovaných periférií



Obr. 43: Popis konektorů pro připojení snímaných periférií

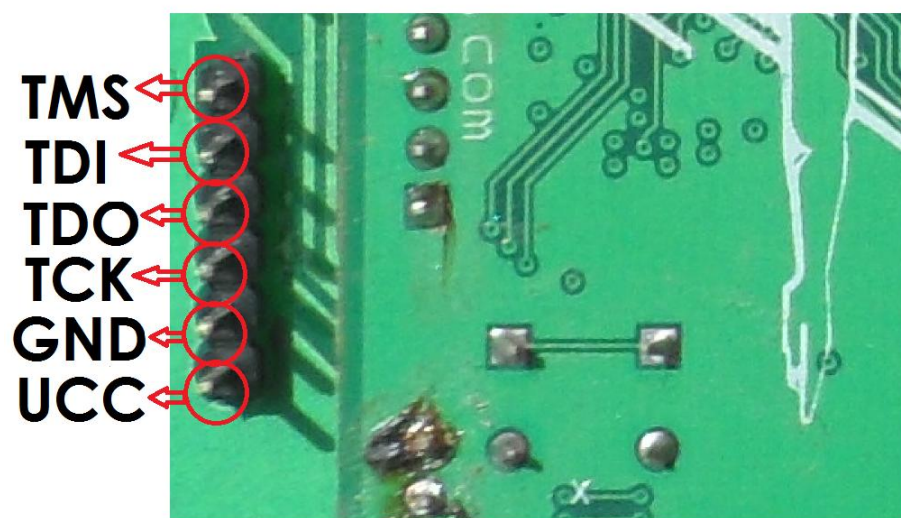
## 7.2 Popis ostatních konektorů

Kvůli chybě v návrhu obvodu FT232, který měl sloužit pro komunikaci s osobním počítačem, musel být využit konektor, který slouží ke komunikaci pomocí standardu UART. Na tento konektor byl připojen zakoupený modul, který již v sobě obsahoval zmiňovaný obvod FT232 a bylo možné ho připojit přímo na svorky RXD, TXD a GND připraveného konektoru. Tím lze komunikovat s osobním počítačem stejným způsobem.



Obr. 44: Konektor RESET, konektory s napájecím napětím a konektory pro komunikaci s osobním počítačem

Poslední zobrazené konektory slouží ke komunikaci obvodu FPGA pomocí rozhraní JTAG. Popis tohoto konektoru je na obr. 45.



Obr. 45: Konektor rozhraní JTAG pro obvod FPGA

## 8 Porovnání dosažených parametrů

Vytvořený logický analyzátor byl porovnán se dvěma komerčně dostupnými analyzátory a to: ASIX SIGMA2 a LOGIC ANALYZER od firmy SALEAE. Jelikož ani jeden ze zmíněných analyzátorů nevlastní a neznám jejich softwarovou výbavu, vychází toto srovnání pouze z tabulkových parametrů jednotlivých výrobců.

### 8.1 Vytvořený analyzátor

Tab. 15: Shrnutí parametrů vytvořeného analyzátoru

Parametr	Hodnota
Maximální vzorkovací frekvence	100 MHz – bez omezení
Synchronní snímací režimy	15bitová sběrnice, I <sup>2</sup> C, SPI
Asynchronní snímací režimy	16bitová sběrnice , RS232, RS485, UART
Maximální počet dat	2048
Možnost ovládání bez použití osobního počítače	ANO
Možnost zobrazení dat bez použití osobního počítače	ANO mimo asynchronního IN-16 režimu
Maximální možný zaznamenaný časový úsek	2m 14s

### 8.2 AXIS SIGMA2

Tab. 16: Shrnutí parametrů porovnávaného analyzátoru AXIS SIGMA2, převzato z [25]

Parametr	Hodnota
Maximální vzorkovací frekvence	200 MHz – 4 vstupy, 100 MHz – 8 vstupů, 50 MHz 16 vstupů
Synchronní snímací režimy	15bitová sběrnice, SPI, I2C
Asynchronní snímací režimy	15bitová sběrnice, UART
Maximální počet dat	14 680 064
Možnost ovládání bez použití osobního počítače	NE
Možnost zobrazení dat bez použití osobního počítače	NE
Maximální možný zaznamenaný časový úsek	Více než 45min



### 8.3 LOGIC ANALYZER SALEAE

Tab. 17: Shrnutí parametrů porovnávaného analyzátoru SALEAE, převzato z [26]

Parametr	Hodnota
Maximální vzorkovací frekvence	24 MHz
Synchronní snímací režimy	8bitová sběrnice, SPI, I2C, I2S
Asynchronní snímací režimy	8bitová sběrnice, UART, PCM, Manchester
Maximální počet dat	10 000 000 000
Možnost ovládání bez použití osobního počítače	NE
Možnost zobrazení dat bez použití osobního počítače	NE

## 9 Závěr

V této práci je popsána základní koncepce univerzálního logického analyzátoru založeného na obvodu FPGA, který je použit v kombinaci s mikrokontrolérem ATmega32. Tato koncepce byla zvolena na základě potřeby snímání sběrnic s vysokou komunikační rychlostí s ohledem na nízkou cenu a možnostmi připojení jednotlivých periférií. Obvod FPGA je použit pro snímání a ukládání stavů na sběrnicích a k přenosu naměřených dat do mikrokontroléru. Ten pak tyto data dále zpracovává a buď je provedeno zobrazení na jednobarevném grafickém displeji, nebo je předá k dalšímu zpracování osobnímu počítači. Pro interpretaci naměřených stavů v osobním počítači byla vytvořena aplikace ve vývojovém prostředí C++ Builder, kterou lze analyzátor také ovládat. Ovládání lze také provádět pomocí dotykové fólie, která je umístěna na vestavném displeji. Toto řešení je využito především při absenci použití osobního počítače.

Vytvořený analyzátor disponuje nyní vzorkovací frekvencí 100 MHz. Tato vzorkovací frekvence není nijak omezena podle počtu použitých pinů. Vnitřní paměť pro zachycená data dokáže uložit 2048 nasnímaných hodnot, to znamená že, nejvyšší zaznamenaný časový úsek je 2 minuty a 14 sekund. Analyzátor umožňuje snímání standardů RS232, RS485, I<sup>2</sup>C, SPI a snímání 16bitové sběrnice. Aplikace disponuje funkcemi pro ukládání a nahrávání hodnot, tisk, generování zdrojového kódu pro mikrokontrolér AVR, základní měření pomocí kurzorů, několik druhů zobrazení a také různé možnosti spouštění.

V dohledné době jsou v úvaze některá zlepšení vytvořeného analyzátoru. Jako první plánovaná změna je zvýšení vzorkovací frekvence na hodnotu blížíící se 200 MHz. Jako druhé vylepšení by se měla osadit součástkami část ke snímání režimu CAN a doplnit také softwarové řešení pro tento režim. Dále by bylo vhodné na volné piny obvodu FPGA připojit externí paměť, která by sloužila k možnosti uložení delšího časového úseku. Poslední plánovou změnou je vytvoření druhé desky plošných spojů, kde by byl pouze vývojový kit FPGA a obvod pro změnu vstupního napětí 16bitové sběrnice. Ostatní okolní periférie pro snímání režimů RS232, RS485 apod., grafický displej a mikrokontrolér ATmega32 by byly vynechány. Tím by se dala tato verze použít pouze jako analyzátor 16bitové sběrnice a režimu UART, ale získali bychom větší kompatibilitu a menší náklady na zhotovení.

# Literatura

- [1] ŠŤASTNÝ, J. *FPGA prakticky*. Praha: BEN – technická literatura, 2010.
- [2] KOLOUCH, J. *Programovatelné logické obvody*. Elektronické skriptum. Brno: FEKT VUT v Brně, 2009.
- [3] MATOUŠEK, D. *Práce s mikrokontroléry ATMELE AVR Atmega16 4.díl*. Praha: BEN – technická literatura, 2006.
- [4] HANTOUCHUSA. *How it works: 4-Wire Analog-Resistive Touch Screens*. [online] citováno 7. dubna 2012. Dostupné na [www](http://www.rlocman.ru/i/File/2010/05/05/1.pdf):  
<http://www.rlocman.ru/i/File/2010/05/05/1.pdf>
- [5] FRÝZA, T. *Mikroprocesorová technika a embedded systémy*. Elektronické skriptum. Brno: FEKT VUT v Brně, 2010.
- [6] MATOUŠEK, David. *Práce s mikrokontroléry ATMELE AT89C2051*. Praha: BEN – technická literatura, 2004. 264 s.
- [7] Serial Peripheral Interface Bus [online]. poslední aktualizace 26. dubna 2012 20:25 [cit. 26. dubna 2012], Wikipedie. Dostupné na [www](http://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus):  
[http://en.wikipedia.org/wiki/Serial\\_Peripheral\\_Interface\\_Bus](http://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus)
- [8] FRÝZA, Tomáš., FEDRA, Zbyněk., ŠEBESTA, Jiří. *Mikroprocesorová technika. Počítačová cvičení. Elektronické skriptum*. Brno: FEKT VUT v Brně, 2009.
- [9] PINKER, J., POUPA, M. *Číslicové systémy a jazyk VHDL*. Praha: BEN – technická literatura, 2009.
- [10] TEXAS INSTRUMENTS. *SN74ALVC164245 16-BIT 2.5-V TO 3.3-V/3.3-V TO 5-V LEVEL-SHIFTING TRANSCEIVER WITH 3-STATE OUTPUTS*. [online] citováno 25. Listopadu 2012. Dostupné na [www](http://www.ti.com/lit/ds/symlink/sn74alvc164245.pdf):  
<http://www.ti.com/lit/ds/symlink/sn74alvc164245.pdf>
- [11] WEBENCH® Design Center [online]. [cit. 26. dubna 2012], TEXAS INSTRUMENTS. Dostupné na [www](http://www.ti.com/ww/en/analog/webench/index.shtml?DCMP=sva_webench&HQS=webench):  
[http://www.ti.com/ww/en/analog/webench/index.shtml?DCMP=sva\\_webench&HQS=webench](http://www.ti.com/ww/en/analog/webench/index.shtml?DCMP=sva_webench&HQS=webench)
- [12] SIPEX. *SP3222E/3232E*. [online] citováno 25. Listopadu 2012. Dostupné na [www](http://www.farnell.com/datasheets/70989.pdf):  
<http://www.farnell.com/datasheets/70989.pdf>
- [13] ANALOG DEVICE. *3.3 V Slew Rate Limited, Half Duplex RS-485/RS-422 Transceiver*. [online] citováno 25. Listopadu 2012. Dostupné na [www](http://www.analog.com/static/imported-files/data_sheets/ADM3493.pdf):  
[http://www.analog.com/static/imported-files/data\\_sheets/ADM3493.pdf](http://www.analog.com/static/imported-files/data_sheets/ADM3493.pdf)
- [14] PHILIPS. *TJA1040 High speed CAN transceiver*. [online] citováno 25. Listopadu 2012. Dostupné na [www](http://www.nxp.com/documents/data_sheet/TJA1040.pdf):  
[http://www.nxp.com/documents/data\\_sheet/TJA1040.pdf](http://www.nxp.com/documents/data_sheet/TJA1040.pdf)
- [15] XILINX. *Xilinx Spartan-3E FPGA Sample Pack User's Guide*. [online] citováno 25. Listopadu 2012. Dostupné na [www](http://www.xilinx.com/products/boards/s3esamplepack/files/S3Euserguide.pdf):  
<http://www.xilinx.com/products/boards/s3esamplepack/files/S3Euserguide.pdf>

- [16] XILINX. *Spartan-3E FPGA Family Data Sheet*. [online]  
citováno 25. Listopadu 2012. Dostupné na www:  
[http://www.xilinx.com/support/documentation/data\\_sheets/ds312.pdf](http://www.xilinx.com/support/documentation/data_sheets/ds312.pdf)
- [17] FEDRA, Zbyněk. *LCD displeje*. [online]  
citováno 25. Listopadu 2012. Dostupné na www:  
[https://krel.feec.vutbr.cz/VYUKA/M\\_EST/MMIA/Texty/Predn6/pr6-artic.pdf](https://krel.feec.vutbr.cz/VYUKA/M_EST/MMIA/Texty/Predn6/pr6-artic.pdf)
- [18] Controlling graphical 128x64 LCD based on KS0108 [online].  
poslední aktualizace 20. července 2007 [cit. 25. listopadu 2012], ScienceProg.  
Dostupné na www:  
<http://www.scienceprog.com/controlling-graphical-128x64-lcd-based-on-ks0108/>
- [19] FTDI CHIP. *Future Technology Devices International Ltd. FT232R USB UART IC*.  
[online] citováno 25. Listopadu 2012. Dostupné na www:  
<http://www.farnell.com/datasheets/1647396.pdf>
- [20] HW server představuje – Sériová linka RS-232[online].  
poslední aktualizace 12. prosince 2005 00:00  
[cit. 25. listopadu 2012], HW. Dostupné na www:  
<http://www.hw.cz/rozhrani/hw-server-predstavuje-seriova-linka-rs-232.html>
- [21] RS485 & 422 [online]. poslední aktualizace 15. leden 1998 11:46  
[cit. 25. listopadu 2012], HW. Dostupné na www:  
<http://www.hw.cz/teorie-a-praxe/dokumentace/rs-485-422.html>
- [22] TEXAS INSTRUMENTS. *LM2596 SIMPLE SWITCHER Power Converter 150kHz 3A Step-Down Voltage Regulator*. [online] citováno 25. Listopadu 2012. Dostupné na  
www:  
<http://www.ti.com/lit/ds/symlink/lm2596.pdf>
- [23] TEXAS INSTRUMENTS. *LM1117-N/LM1117I 800mA Low-Dropout Linear Regulator*. [online] citováno 25. Listopadu 2012. Dostupné na www:  
<http://www.ti.com/lit/ds/symlink/lm1117-n.pdf>
- [24] KADLEC, V. *Učíme se programovat v C++ BUILDERU a jazyce C++*. Brno:  
Computer Press, 2004.
- [25] SIGMA2 [online]. poslední aktualizace 3. prosince 2012 [cit. 4. března 2013], ASIX.  
Dostupné na www: [http://www.asix.cz/dbg\\_sigma.htm](http://www.asix.cz/dbg_sigma.htm)
- [26] Logic [online]. poslední aktualizace 2013 [cit. 4. března 2013], SALEAE.  
Dostupné na www: <http://www.saleae.com/logic>

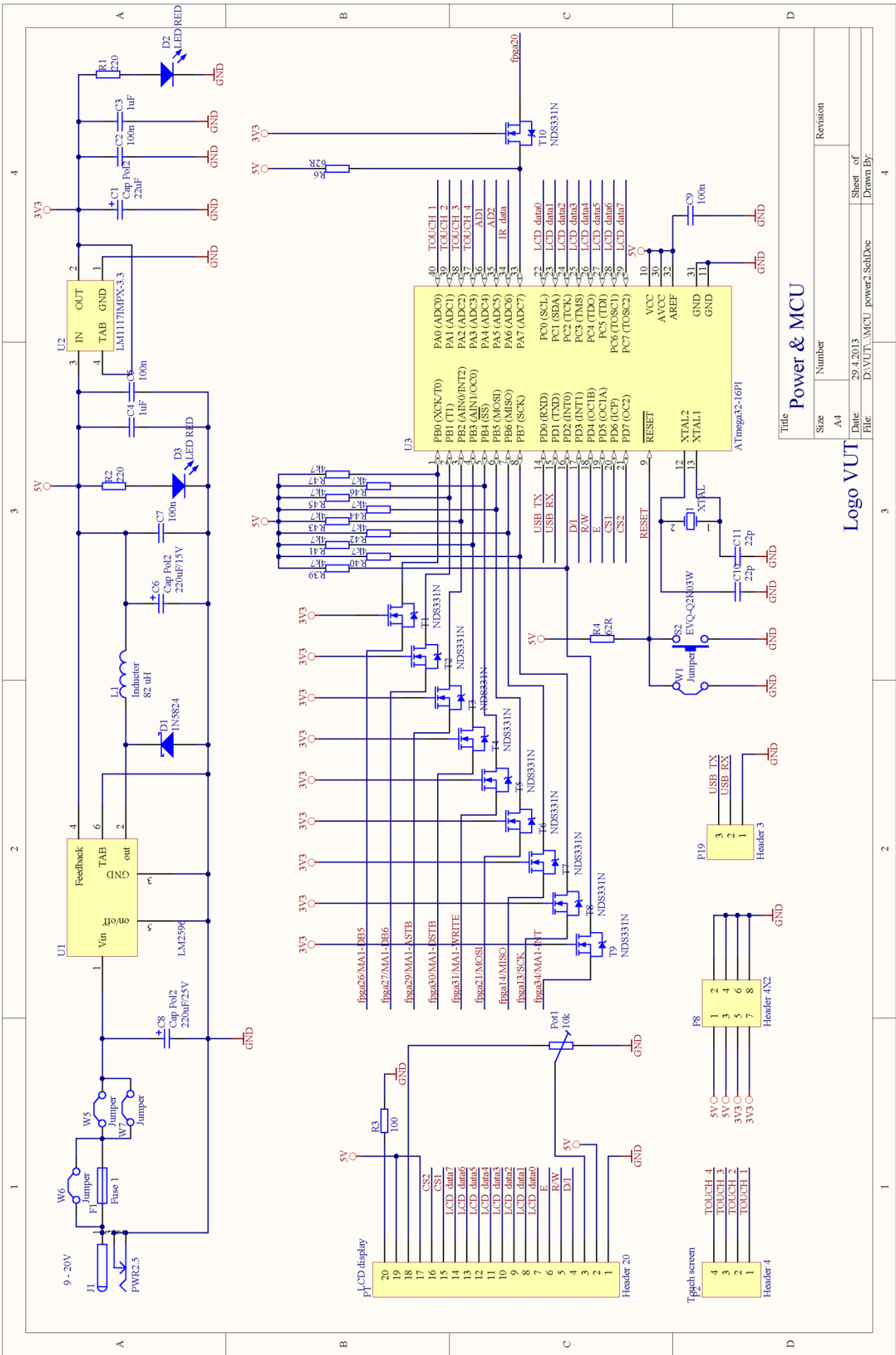
# Seznam zkratek a symbolů

FPGA	Field Programmable Gate Arrays
USB	Universal Serial Bus
I <sup>2</sup> C	Inter-Integrated Circuit
SPI	Serial Peripheral Interface
PLD	Programmable logic device
DSP	Digital Signal Processor
MIPS	Milions of Instruction Per Second
ALU	Arithmetic Logic Unit
MAC	Multiply–accumulate
MMACs	Milions of Multiply-and-ACcumulate operations per Second
A/D	Analog-to-Digital
IrDA	Infrared Data Association
PC	Personal computer
UART	Universal Asynchronous Receiver/Transmitter
ASIC	Application-specific integrated circuit
CLB	Configurable Logic Block
LUT	Look-Up Table
CMOS	Complementary metal–oxide–semiconductor
TTL	Transistor–transistor logic
LVTTL	Low-voltage Transistor–transistor logic
PCI	Peripheral Component Interconnect
RAM	Random-access memory
MOSFET	Metal–oxide–semiconductor field-effect transistor
USART	Universal synchronous/asynchronous receiver/transmitter
LCD	Liquid crystal display
PET	Polyetylen
ITO	Indium Tin Oxid
MISO	Master Input, Slave Output
SS	Slave Select
MOSI	Master Output, Slave Input

VHSIC	Very High Speed Integrated Circuits
IEEE	Institute of Electrical and Electronics Engineers
VHDL	VHSIC hardware description language

# SEZNAM PŘÍLOH

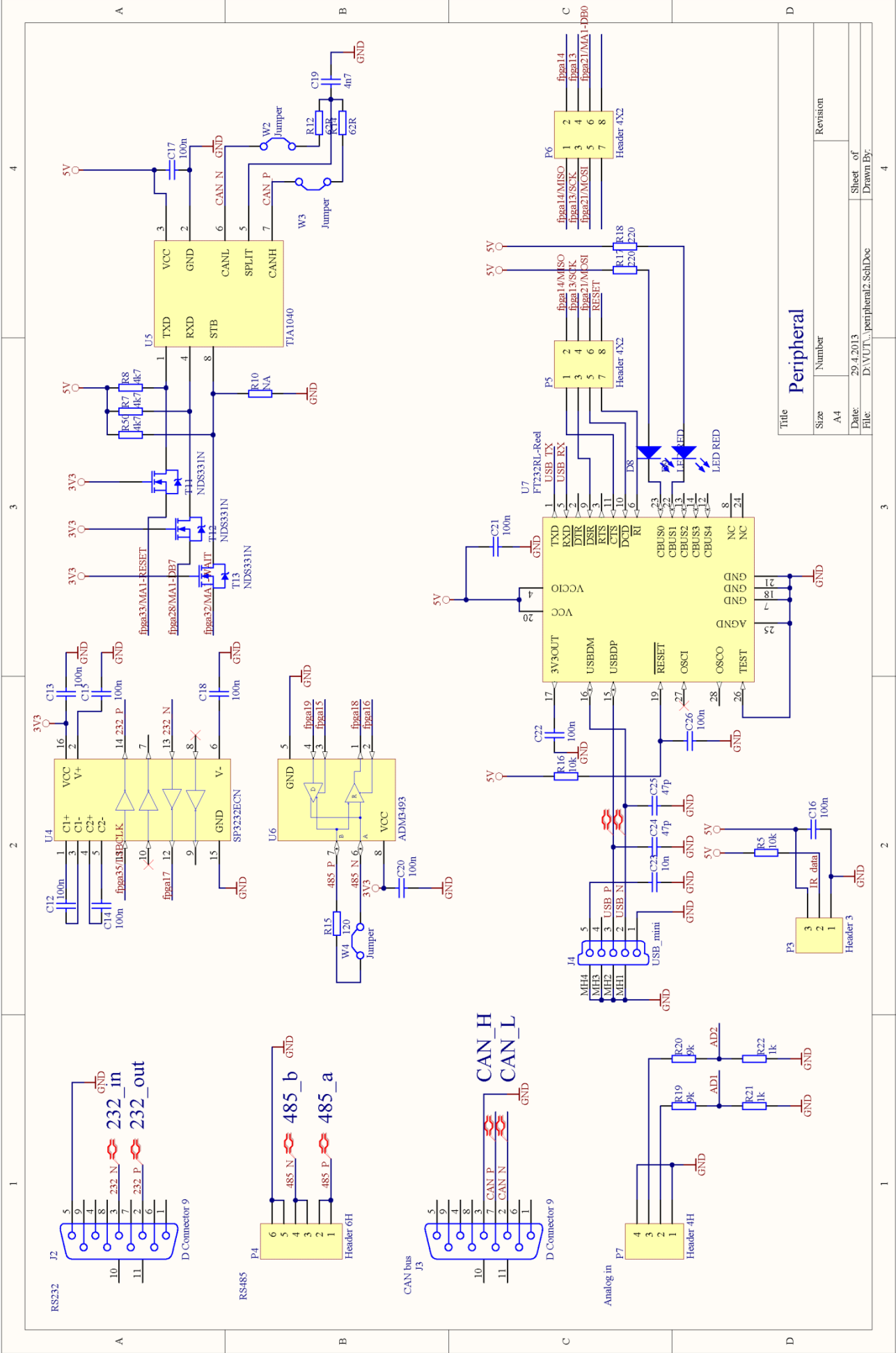
Příloha 1: Schematické zapojení napájecí části a mikrokontroléru ATmega32 .....	61
Příloha 2: Schematické zapojení periférií .....	62
Příloha 3: Schematické zapojení GPIO sběrnice .....	63
Příloha 4: Schematické zapojení připojení FPGA obvodu .....	64
Příloha 5: Vytvořené blokové schéma pro obvod FPGA .....	65
Příloha 6: Deska plošných spojů strana TOP – zobrazení není v měřítku 1:1 .....	66
Příloha 7: Deska plošných spojů strana BOTTOM – zobrazení není v měřítku 1:1 .....	67
Příloha 8: Vytvořeny analyzátor .....	68
Příloha 9: Vytvořená počítačová aplikace .....	69
Příloha 10: Seznam použitých součástek .....	71



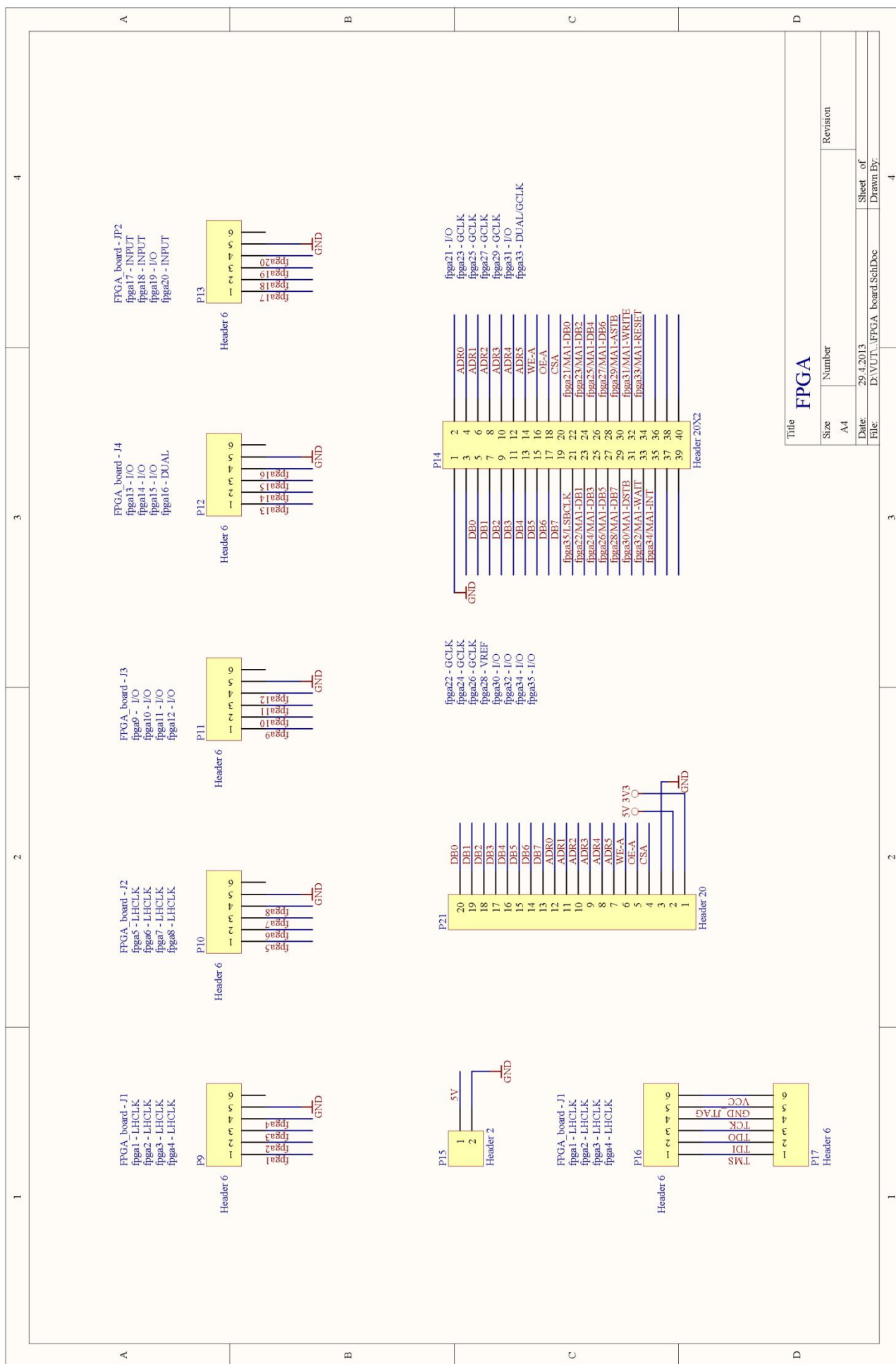
Příloha 1: Schematické zapojení napájecí části a mikrokontroléru ATmega32



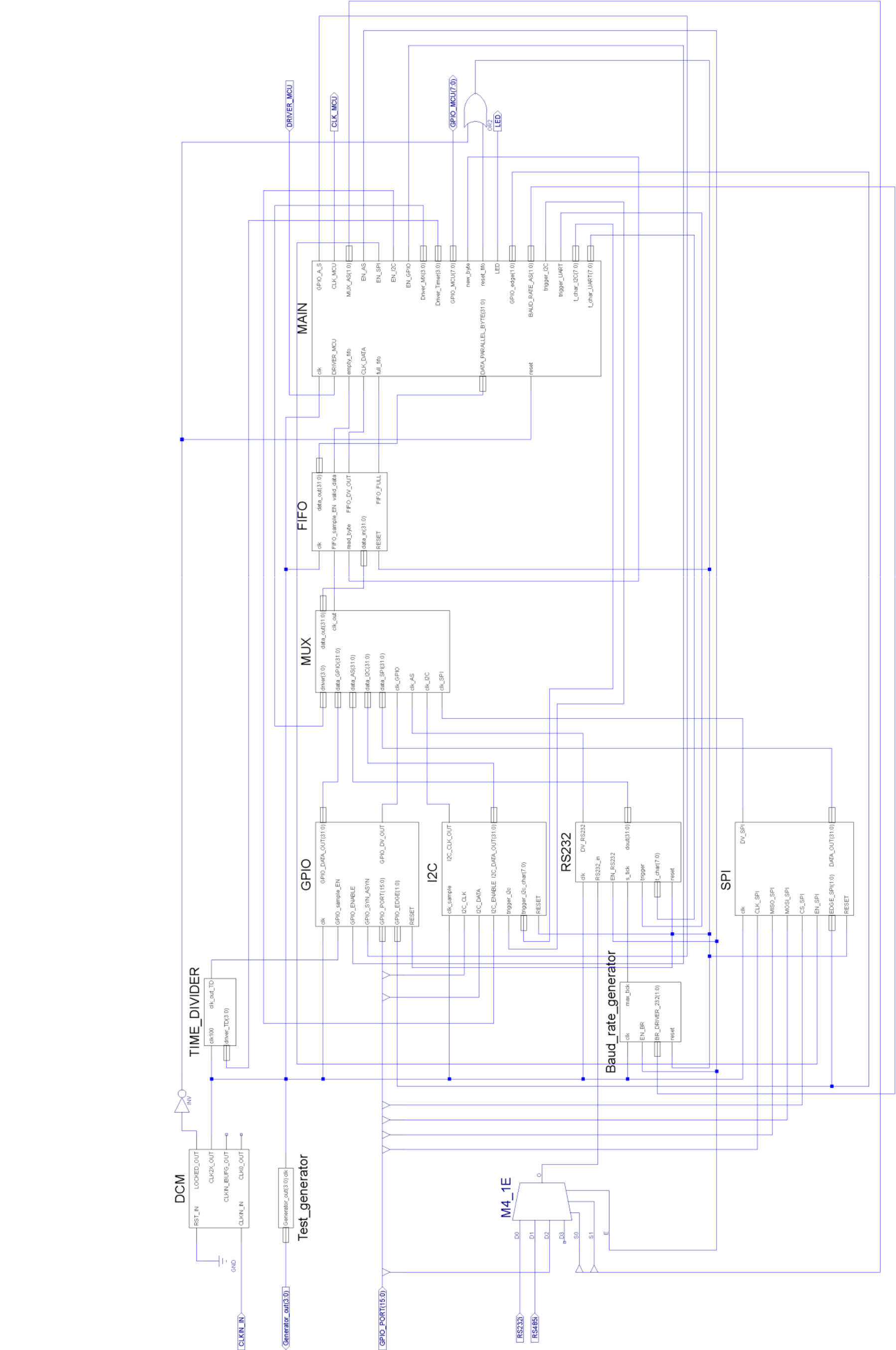
Příloha 2: Schematické zapojení periférií



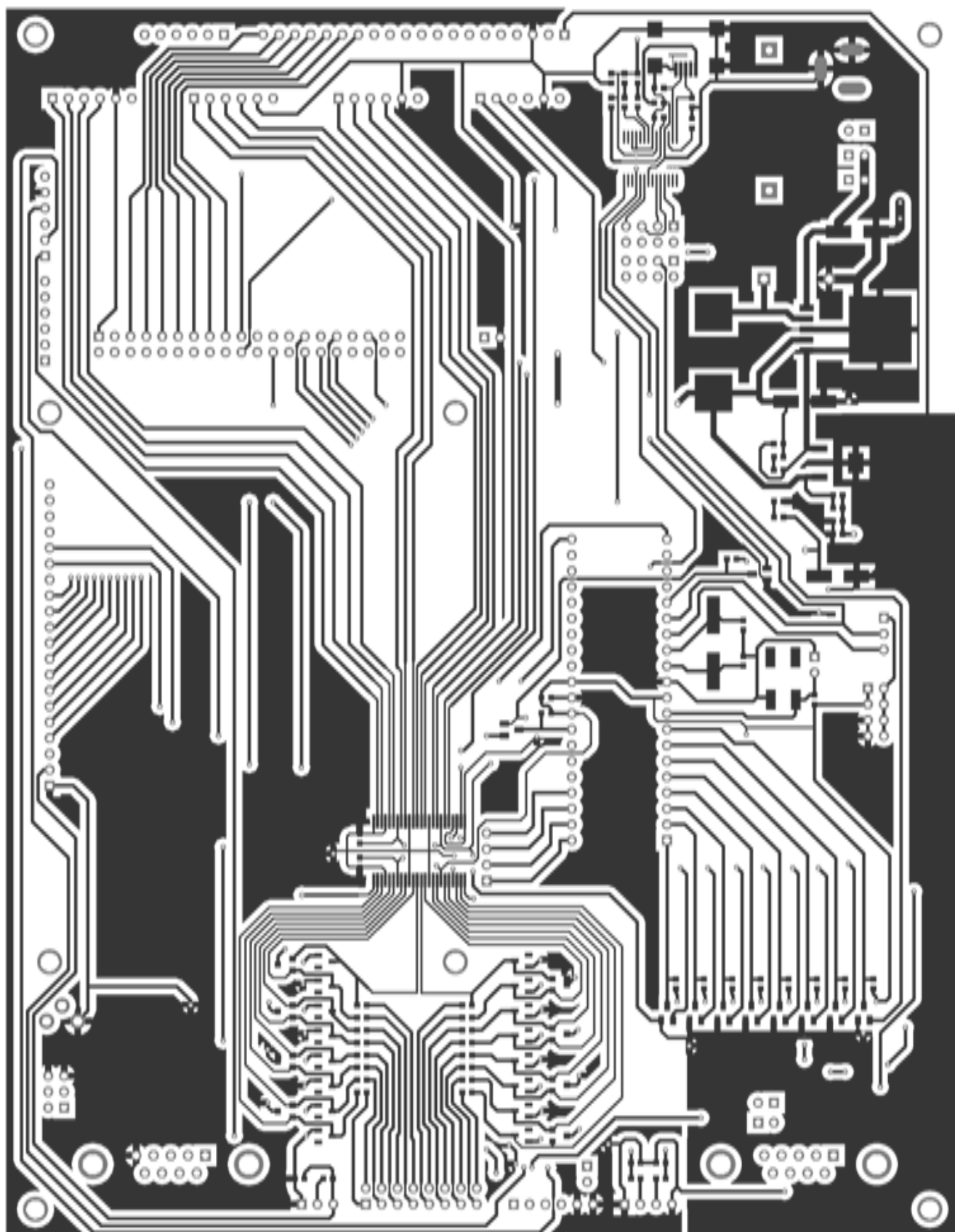




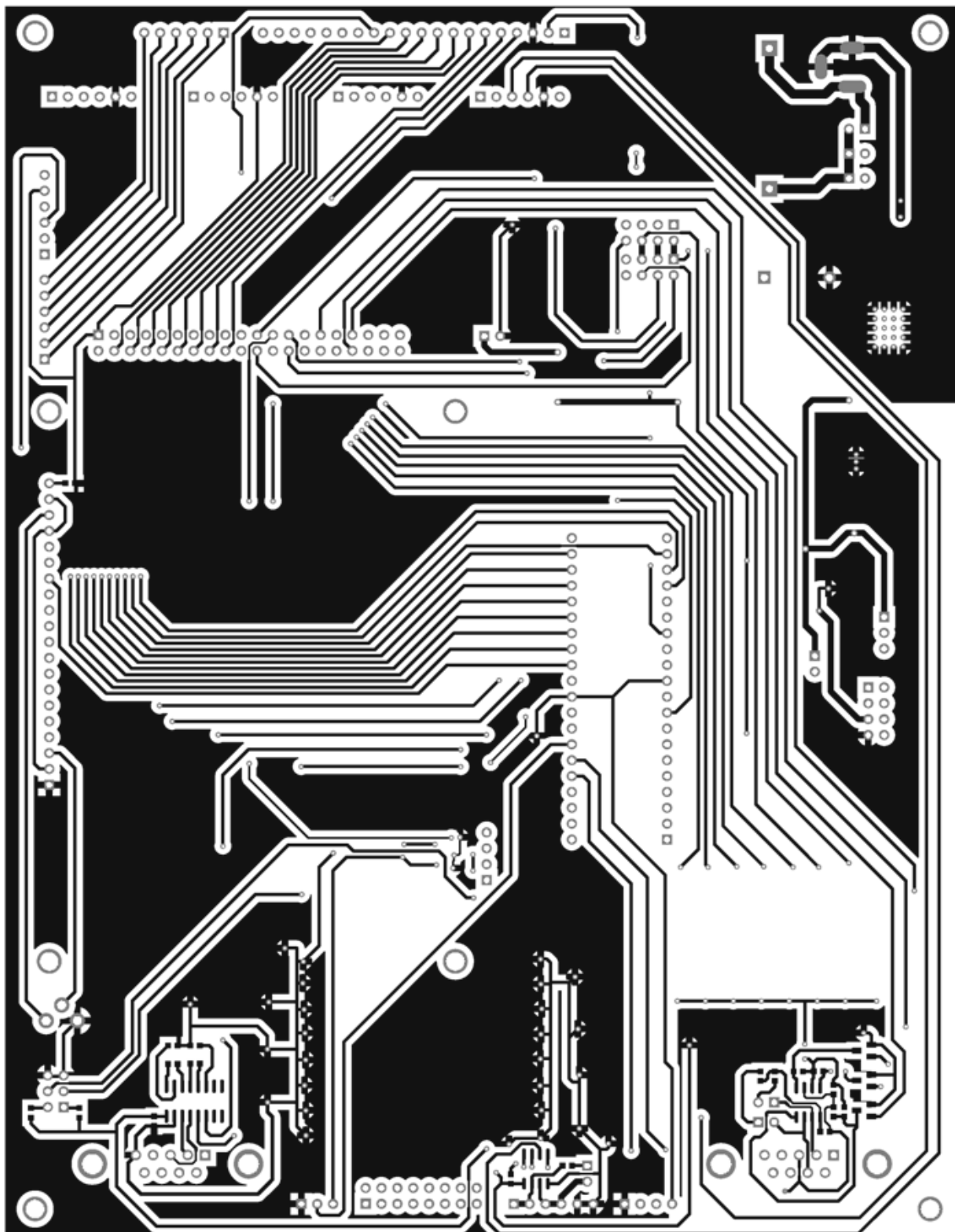
Příloha 4: Schematické zapojení připojení FPGA obvodu



Příloha 5: Vytvořené blokové schéma pro obvod FPGA



Příloha 6: Deska plošných spojů strana TOP – zobrazení není v měřítku 1:1

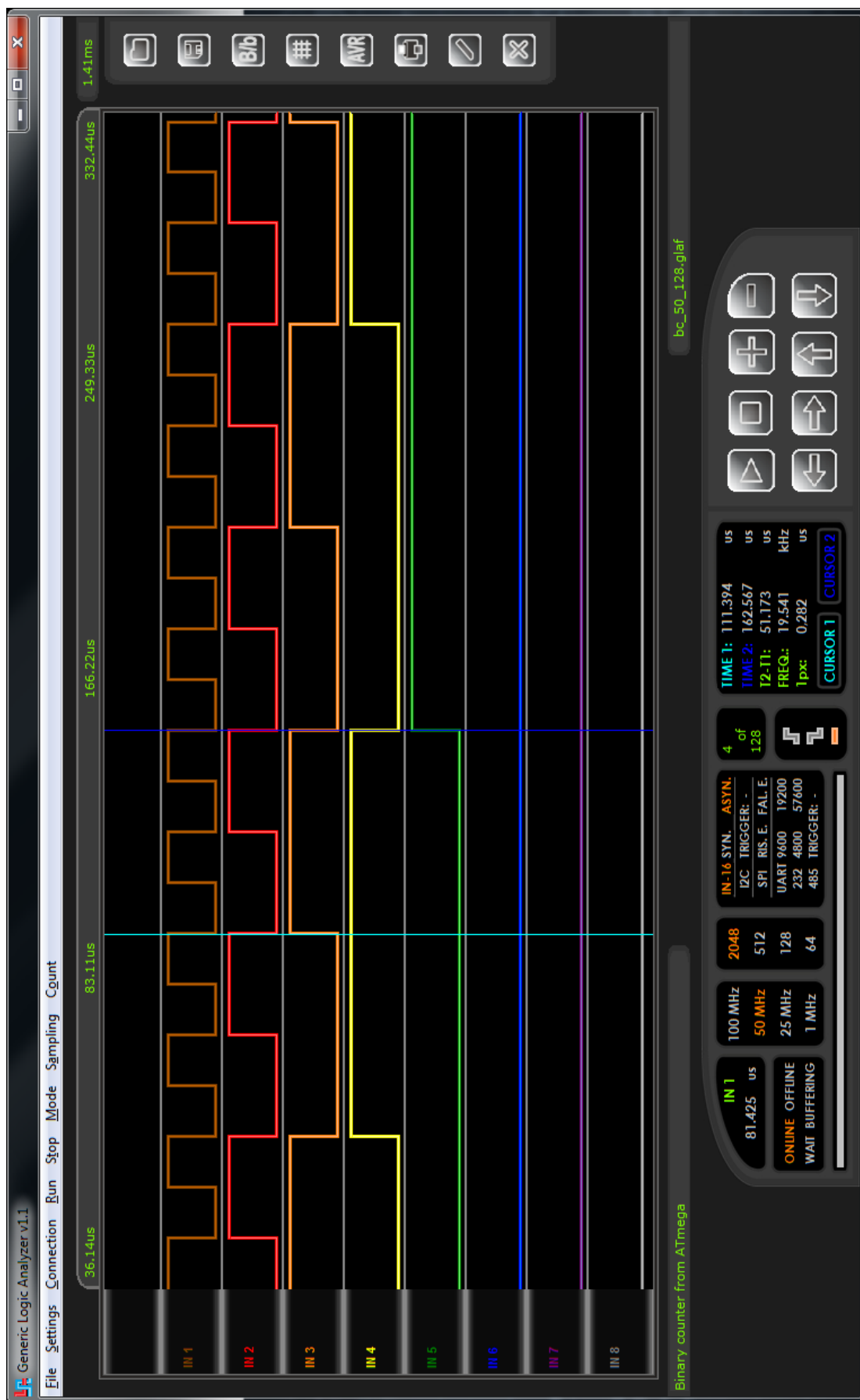


Příloha 7: Deska plošných spojů strana BOTTOM – zobrazení není v měřítku 1:1





Příloha 8: Vytvořený analyzátor



Příloha 9: Vytvořená počítačová aplikace



### Seznam použitých součástek:

Součástka	Hodnota	Pouzdro	Počet
<b>XTAL</b>	16 MHz	HC49/US	1
<b>C1, C6, C8</b>	220μF/25V	8x10,5mm	3
<b>C2, C5, C7, C9, C12, C13, C14, C15, C16, C17, C18, C20, C21, C22, C26, C27, C28</b>	100n	C0603	17
<b>C3, C4</b>	1uF	C0603	2
<b>C10, C11</b>	22p	C0603	2
<b>C19</b>	4n7	C0603	1
<b>C23</b>	10n	C0603	1
<b>C24, C25</b>	47p	C0603	2
<b>D1</b>	1N5824	DO-204AL	1
<b>D2, D3, D8, D9</b>	LED RED	D0603	4
<b>D10, D11, D12, D13, D14, D15, D16, D17, D18, D19, D20, D21, D22, D23, D24, D25</b>	BAS70	SOT-23_N	16
<b>F1</b>	0,5A	5x20mm	1
<b>J1</b>	PWR	KLD-0202	1
<b>J2, J3</b>	D Connector 9	DSUB9	2
<b>J4</b>	USB_mini	USB_mini	1
<b>L1</b>	82μH	8x12,8mm	1
<b>P1, P21</b>	Header 20	HDR1X20	2
<b>P2</b>	Header 4	HDR1X4	1
<b>P3</b>	Header 3	HDR1X3H	1
<b>P4</b>	Header 6H	HDR1X6H	1
<b>P5, P6, P8</b>	Header 4X2	HDR2X4	3
<b>P7</b>	Header 4H	HDR1X4H	1
<b>P9, P10, P11, P12, P13, P16, P17</b>	Header 6	HDR1X6	7
<b>P14</b>	Header 20X2	HDR2X20	1
<b>P15</b>	Header 2	HDR1X2	1

<b>P18</b>	Header 3X2	HDR2X3	1
<b>P19</b>	Header 3	HDR1X3	1
<b>P20</b>	Header 8X2H	HDR2X8H	1
<b>Pot1</b>	10k	trimr	1
<b>R1, R2, R17, R18</b>	220	R0603	4
<b>R3</b>	100	R0603	1
<b>R4, R6, R12, R14</b>	62R	R0603	4
<b>R5, R16, R48, R49</b>	10k	R0603	4
<b>R7, R8, R39, R40, R41, R42, R43, R44, R45, R46, R47, R50</b>	4k7	R0603	12
<b>R10</b>	NA	R0603	1
<b>R15</b>	120	R0603	1
<b>R19, R20</b>	9k	R0603	2
<b>R21, R22</b>	1k	R0603	2
<b>R23, R24, R25, R26, R27, R28, R29, R30, R31, R32, R33, R34, R35, R36, R37, R38</b>	1k5	R0603	16
<b>S2</b>	B3S-1002	6x4,3mm	1
<b>T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13</b>	NDS331N	SSOT-3	13
<b>TVS1, TVS2, TVS3, TVS4, TVS5, TVS6, TVS7, TVS8</b>	SP0502BAJTG	SOT-23	8
<b>U1</b>	LM2596	TO-263_2	1
<b>U2</b>	LM1117IMPX-3.3	MP04A_N	1
<b>U3</b>	ATmega32-16PI	40P6	1
<b>U4</b>	SP3232ECN	NSO16_N	1
<b>U5</b>	TJA1040	NSO8_L	1
<b>U6</b>	ADM3493	NSO8_L	1
<b>U7</b>	FT232RL-Reel	SSOP-28_N	1
<b>U8</b>	SN74ALVC164245DL	DL048	1
<b>W1, W2, W3, W4, W5, W6, W7</b>	Jumper	HDR1X2	7

Příloha 10: Seznam použitých součástek